The Dissertation Committee for Calvin Tsay
certifies that this is the approved version of the following dissertation:

Modeling and Optimization of Process Systems for Unconventional Technologies and Feedstocks

Committee:

Michael Baldea, Supervisor

George Biros

Thomas F. Edgar

Gary T. Rochelle

Modeling and Optimization of Process Systems for Unconventional Technologies and Feedstocks

by

Calvin Tsay

DISSERTATION

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN

May 2020

Dedicated to my family, Shirley, Frank, and Jerry.

Acknowledgments

I would firstly like to thank my advisor, Dr. Michael Baldea, who has been my strong supporter and mentor throughout the ups and downs of graduate school. In addition to academic guidance, his mentorship has taught me to be a more efficient worker, a more critical thinker, and a better writer. Thank you for always supporting your students and encouraging us to seek opportunities far and wide.

My committee members, Professors George Biros, Thomas Edgar, and Gary Rochelle, were incredibly helpful sources of feedback, advice, and mentoring. I was fortunate to learn from and collaborate with all of them. Professor Mark Stadtherr joined UT later on and also provided valuable feedback and collaboration in process optimization.

My graduate studies would be impossible without Drs. Francisco Vargas and Mohan Boggara, who mentored me as an undergraduate researcher at Rice and encouraged me to continue as a graduate student. I also thank Dr. Yu Luo for teaching me about PSE.

Dr. Richard Pattison mentored and trained me for my first two years as a graduate student. I am extremely thankful for his patience and dedication in teaching me many PSE concepts. I am also greateful to have had the opportunity to participate in the Innovation Corps program with Richard, and for the entrepreneurial mentorship we received from Michael Piana. I am further grateful to all my friends in industry who helped with our study.

Moreover, I thank all of the industrial collaborators I worked with through the TWCCC, especially Drs. Jesus Flores-Cerrillo, Ankur Kumar, and Jun Shi at Linde (formerly Praxair). Dr. Zheng Li gave me the opportunity to study data science at Genentech. He and Dr. Victor Saucedo mentored me during the internship. I also thank Drs. Steve

Hodson, Rob Johnson, and Ben Weinstein at P&G for the collaboration. These industrial perspectives, data, and constant feedback provided valuable directions for my research.

Professor Alexander Mitsos hosted me as a visiting researcher at RWTH Aachen University. I greatly enjoyed collaborating with his group during and after my visit.

Dr. Ruth Misener at Imperial College London helped me hone my ideas into a fellowship application and guided me in preparing for interviews. Drs. Baldea and Pattison, and Professor Costas Pantelides read and provided great feedback on my applications. I am extremely excited to be joining the Computational Optimisation Group at Imperial.

During my time at UT, I have enjoyed the privilege to work alongside many PSE researchers, including Richard, Cara, Siyun, Conan, Ray, Matt, Abby, Ankur, Corey, Melissa, Hari, Jodie, Joannah, Joe, Morgan, Lingqing, Omar, Giannis, Fernando, Jun, and Alkis. I especially thank Jodie—who shared an office with me for five years—for the helpful discussions, Ray for showing me where to find good food near CPE, and Ankur, Matt, and Richard for continuing to collaborate with me after they graduated.

I am thankful to all my friends and family for their continuous love and belief in me. Oscar, Mike, and Pacha kept me in good spirits throughout graduate school; there was never a dull moment in our house. Moreover, I am grateful for my friends in Austin (esp. Ben, Olivia, ECC community) and beyond (esp. Thomas, Akshay, Jennifer) who were always encouraging and invested in my progress. I apologize for not keeping in touch better all these years! I am very thankful to Ariel, who has been a consistent source of wisdom, encouragement, and inspiration. Finally, I will always be thankful to my family, Shirley, Frank, and Jerry, for loving and supporting me from the beginning. My dissertation is dedicated to them.

Modeling and Optimization of Process Systems for Unconventional Technologies and Feedstocks

Calvin Tsay, Ph.D.
The University of Texas at Austin, 2020

Supervisor: Michael Baldea

In the present era, the petrochemical/chemical process industries must adapt to unconventional feedstocks and energy sources, in order to keep pace with increased competition, regulatory pressure, and changing markets. However, developing processes compatible with these changes requires deviating from traditional and accepted process design and operation paradigms. This dissertation addresses fundamental challenges related to this transition from three angles: incorporation of custom (and detailed) models into process design, integration of variable operation with process design, and optimization of transient process operations.

The first part of the dissertation introduces a framework for modeling, simulation, and optimization of process flowsheets incorporating highly detailed physical models of important and complex process units, termed "multi-resolution flowsheets". The framework relies on pseudo-transient continuation as a numerical method and allows for the robust optimization of large-scale process models. Several case studies demonstrate the method, including process flowsheets featuring both intensified (e.g., dividing-wall distillation column, multistream heat exchanger) and unconventional (e.g., quenched reactor, packed column

for carbon capture) process units. Furthermore, these results reveal significant benefits of considering the added level of detail at the design stage. Finally, an avenue is presented to accelerate the convergence of the pseudo-transient method, which is especially important for the large-scale models considered.

In the second part of the dissertation, the focus shifts to process design optimization for variable operation, or optimization under uncertainty. Here, I present a method for process design that considers the effect of uncertain physical parameters (assumed to follow continuous probability distributions), using a formulation that exploits the semi-infinite nature of dynamic optimization. I compare the method to traditional "scenario-based" approaches using both theoretical analyses and multiple case studies. In addition to demonstrating the effectiveness of the proposed method, these case studies also emphasize the importance of considering several practically relevant uncertainties during process design.

The final part of the dissertation examines explicit consideration of process dynamics for operational optimization. First, I examine periodic (dynamically intensified) processes, which operate at a cyclic steady state. I present a pseudo-transient method for robust optimization of fully discretized dynamic process models, and I present an approach for implementing cyclic conditions based on their fundamental relation to material/energy recycle loops. Lastly, I propose a framework for optimal production scheduling in fast changing market situations. Towards this end, I show how data-driven dynamic models can represent the behavior of a set of scheduling-relevant (physical or latent) variables. A method is also given for executing scheduling calculations using these models, and the framework is demonstrated by considering the demand response operation of both simulated and real-world air separation units.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Chemical and petrochemical plants turn raw materials, typically harvested from nature, into important and valuable products via a series of chemical and/or physical transformations. The modern petrochemical and chemical industries are strongly influenced by increased global competition, regulatory pressure, and unconventional feedstocks and energy sources [171]. As a result, these industries seek to adopt new process technologies that improve economic performance, increase energy efficiency, and lower the environmental impact of production facilities. The workflow for developing a new process generally comprises (i) idea generation and conceptual design, (ii) detailed modeling, engineering, and design, (iii) pilot plant testing, and (iv) large-scale operations. This workflow is depicted in 1.1.

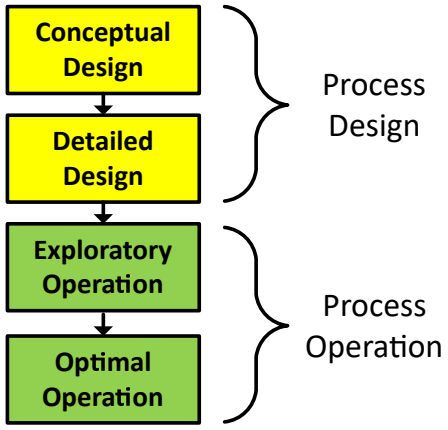

Figure 1.1: Workflow for new process development.

As shown in Figure 1.1, the two broad steps in the process development workflow are process design and process operation. In both these stages, computational tools for process modeling, simulation, and optimization are used to aid in making key decisions.

Process design refers to the determination of the optimal configuration of steps, and corresponding processing units, to produce the desired chemical products. Chemical processes typically comprise numerous unit operations that fulfill specific functions, such as forming products through chemical reactions, heating/cooling materials, etc. Throughout the various stages of process design, the process is represented as a *flowsheet* model, which combines mathematical models of individual units with equations defining the associated connections and interactions. In this context, *modeling* refers to defining the (steady-state) material and energy balances for a given flowsheet, along with the requisite constitutive relations, and *simulation* covers the (numerical) approaches taken to solve the aforementioned balance equations. Furthermore, *optimization* techniques can be applied to improve process designs at various stages. At the *conceptual* level, the optimal flowsheet configuration is selected from a superstructure (typically based on simplified models), while *detailed* design entails selecting the unit specifications (column staging, reactor dimensions) and operating conditions (pressures, temperatures, flow rates, compositions) for a given, fixed flowsheet structure, in which unit operations are described in as much detail as possible [32, 230].

Process operation refers to the post-commissioning activities through which engineers manipulate the process inputs to run a process in the most economic or efficient manner. Early stage process operations tend to be more exploratory, using pilot-scale processes to generate data to validate/improve mathematical models, further optimize the process design, and design future experiments [171]. Later stage process operations are typically more concerned with economic, environmental, or regulatory objectives. In contrast with the above (steady-state) design decisions, process operation is concerned with time-varying behavior,

and computational tools (e.g., process simulators) require *dynamic* models of processes and units in this context to predict, and optimize, process operations. Optimization tools are employed both in *scheduling*, where the optimal production strategy is determined (typically based on simplified/linear models), and in *control*, where the aforementioned strategy is implemented in real time [18]. For the latter, the use of optimization tools for control is particularly manifest in model predictive control (MPC), which has been widely adopted by industry [11].

Given changes in feedstocks and energy sources, modern chemical processes must often be designed to operate with new technologies (i.e., processing steps), and to participate in unfamiliar economic markets. Therefore, the contributions of this dissertation aim to address challenges in process modeling and optimization related to the advent of unconventional technologies, feedstocks, and operations. In light of the aforementioned economic pressures, this dissertation further aims to meet the goals of "process intensification", broadly defined as *any chemical engineering development that leads to a substantially smaller, cleaner, safer, and more energy efficient technology* [14, 17, 182, 239, 249].

At the advent of the field, **process intensification** was largely concerned with equipment miniaturization, and engineers anticipated orders of magnitude reduction in equipment sizes and capital cost in comparison to conventional equipment of the same capacity. These potential improvements were linked to vast increases in the rates of heat, mass, and momentum transfer in small, modular process units; however, it became clear that based on the definition provided, the scope of the process intensification field is in fact much broader than producing compact (modular) systems. In fact, process intensification presents in many forms, including: novel unit operations (e.g, for carrying out chemical reactions, heat exchange, or separation); multifunctional equipment (e.g., reaction/separation equipment); alternative energy sources (e.g., microwaves for heating); and novel operational strategies

such as periodic operation [17, 249]. The diversity of process intensification technologies is evident through the wide array of applications they have found in the chemical industry.

Focusing on related challenges in process design (i.e., unconventional processing steps, intensified unit designs) and in process operations (i.e., dynamically intensified units, electricity market participation), this dissertation examines computational tools for the modeling and optimization of chemical process systems. I first begin with an industrial survey of current best practices and challenges, regarding computational modeling and optimization, in Chapter 2. The remainder of dissertation roughly follows these challenges through the process development workflow shown in Figure 1.1:

I) In Part I of the dissertation, I discuss the incorporation of custom, detailed models of unit operations into **process design optimization**. Where there has been a considerable amount of literature regarding new and intensified processing steps [249], there are few frameworks for their custom modeling and incorporation into process flowsheet optimization. I provide further motivation for this work in Chapter 3, as well as brief background on relevant numerical methods. Focusing on the canonical reaction/separation/recycle structure, Chapter 4 then introduces a method for process design optimization including detailed physical models of key and/or intensified units, which I will refer to as "multi-resolution" flowsheet optimization. Next the method is presented for multistream heat exchangers (a multi-functional intensified unit) in Chapter 5 and for advanced carbon capture plants (a design for a non-traditional feedstock) in Chapter 6. Finally, Chapter 7 examines acceleration of the numerical techniques used for flowsheet simulation and optimization in Chapters 4–6.

II) In Part II of the dissertation, the focus shifts to optimal process design optimization for variable operation, bridging **both process design and operation**. Specifically,

the incorporation of unconventional feedstock or processing steps often results in considerable uncertainty at the design stage, where the actual operating point is therefore unknown. This part examines the consideration of such potential variations in (steady-state) operating point during process design. First, Chapter 8 provides background on optimization under uncertainty and the scenario-based approaches typically used. In Chapter 9, I describe a novel approach for modeling uncertainty in process flowsheets and demonstrate its computational advantages in stochastic optimization, compared to the techniques reviewed in Chapter 8. The new approach is applied to design three reaction/separation/recycle processes (Chapter 10) and a natural gas liquefaction process (Chapter 11) for variable operating point(s) related to several practical uncertainties.

III) In Part III of the dissertation, I address challenges pertaining to dynamic process intensification and flexible production in **optimal process operation**. I focus on two particular directions: the modeling of cyclic (dynamically intensified) processing units, and the scheduling of production to exploit modern deregulated electricity markets. Chapter 12 provides a brief overview of dynamic modeling and optimization, as well as challenges as they pertain to the above two areas. I then present a framework for the robust simulation and optimization of periodic processes in Chapter 13 and demonstrate its applicability to large-scale industrial proceses (including those with complicated, steady-state processing steps) in Chapter 14. Focusing on modern electricity markets, Chapter 15 presents a framework for the fast scheduling of an industrial air separation process, which consumes electricity as its primary feedstock. Lastly, in Chapter 16 I provide a data mining approach to reduce the amount of manual effort required for the scheduling framework.

Chapter 2

A Survey of Optimal Process Design Capabilities and Practices in the (Petro)Chemical Industries[†]

2.1 Motivation: Identifying Challenges in Process Design Optimization

The petrochemical and chemical industries began developing simulation tools for individual unit operations in the middle-1950s [220], and, while limited in number, successful use of early simulation tools demonstrated important cost savings [102, 176, 221]. Initially, many companies developed and maintained their own proprietary simulation tools. While offering the benefit of fostering deep in-house expertise and retaining complete control over the source code, such tools are typically hard to maintain (particularly when transitioning between computational platforms and programming languages) and require dedicated software engineering departments in addition to the engineers who use them. The latter disadvantages have provided the impetus for the development of general process simulators by specialized commercial entities, an effort that gained momentum in the 1980s and continues to date. Through the years, computer process simulators have become accepted tools for engineering educators and practitioners alike [176, 264], especially at (large) companies able to invest in the purchase, deployment, and personnel training for a commercial, third-party process simulation software.

[†]The contents of this chapter are largely based on the following publication: C. Tsay, R.C. Pattison, M.R. Piana, and M. Baldea. A survey of optimal process design capabilities and practices in the chemical and petrochemical industries. Comput. Chem. Eng., 112:180–189, 2018. C.T. is the primary author of the manuscript.

The process systems engineering research community has given significant attention to advancing capabilities for large-scale process optimization [29, 71], and some of the existing commercial process simulation packages now offer deterministic optimization capabilities. Nevertheless, in many practical situations, process design optimization remains an empirical effort consisting of a (guided) trial-and-error search by an experienced engineer. When performed, computational optimization is often carried out with simplified process models or as a piecewise/fragmented optimization of the flowsheet [264]. Existing literature [29, 78, 100, 176, 232, 259] suggests several potential causes for this discrepancy between research advances, commercial software capabilities, and the situation in practice:

- a lack of clarity concerning when the application of rigorous optimization techniques is appropriate and/or beneficial: uncertainty in system parameters (e.g., physical properties) and/or equipment cost can diminish the practical use of the results of deterministic optimization calculations, particularly at the early stages of conceptual design

- problem definition: the choice of decision variables and constraints is not immediately obvious for a new process and requires expert input even in the case of a more established plant design

- setup and computational time are difficult to predict and may be difficult to accommodate when tight project execution timelines are imposed. As I will discuss later, these computational issues are directly related to the robustness of numerical algorithms and their ability to deal with, e.g., algebraic loops and nonlinear, poorly conditioned systems of equations for which a feasible initial guess is difficult to find.

Motivated by the above, I carried out a cross-industry survey of over one hundred practitioners involved in the area of optimal process design, focusing on current industrial practice,

including design project workflows, techniques and approaches, and the associated issues perceived by workers at different experience levels and in different corporate roles.

This chapter reports my aggregate findings, which are largely technology-agnostic. The chapter begins with a working definition for optimal process design, my initial assumptions, and the interview methodology. These are followed by general findings from the interviews broken down by function, insights specific to particular (petro)chemicals industries, and overarching conclusions. While the interviewee cohort represents a wide sampling of the industrial sectors of interest, the conclusions of this survey result from the opinions of and challenges encountered by the interviewed individuals and may not necessarily reflect the status of the entire industry. The findings reveal a very broad spectrum of optimization use among industries, beginning with the fact that personnel in different industries have vastly different interpretations of the notion of "optimization" and the consequent wide variation in practical applications of computer-aided process optimization tools. The presentation of this chapter follows closely the study published in Tsay et al. [262].

2.2 Background and Description of the Survey Approach

Problem Definition. When creating and evaluating a new (petro)chemical process concept, the best design must be carefully selected from multiple candidate process configurations. Research in this area, termed *process synthesis*, has converged towards representing the candidates as a comprehensive superstructure, with the expectation that the optimal structure is contained within and can be identified via a (mixed-integer) optimization procedure. Many methods have been proposed for systematic generation and optimization of flowsheet superstructures [281], and several industrial success stories demonstrate the large potential benefits of process synthesis techniques [235].

Although considerable developments have been made in optimization and computational capabilities, most methods for optimization-based process synthesis still rely on heuristics and/or decomposition techniques, whereby subsystems of the flowsheet (e.g., heat exchanger networks, distillation systems, reactor networks) are optimized separately [54, 216, 230]. When the full flowsheet is considered, it is impractical, except for simple processes featuring a small number of unit operations, to use detailed and nonlinear process unit models [230], and, as a consequence, optimization-based synthesis problems in the literature are typically formulated as mixed-integer linear programs (MILPs). The interested reader is referred to Chen and Grossmann (2017) [54] for more details on recent progress in *process synthesis* techniques and current challenges.

For the candidate process configuration(s) selected at the *synthesis* stage, engineers then use detailed mathematical models to ensure that the optimal design is selected for manufacturing a given product palette. At this stage, the impact of model accuracy increases significantly, and accurate models are often developed using pilot-scale experiments. I will refer to this important step as process **design optimization**, specifically defining it as *the activity of identifying, for a fixed group of process units and connectivity structure, the process characteristics (unit sizes and internal configurations, stream flow rates, operating pressures, etc.) that maximize an objective function reflecting process economics while satisfying safety, quality, throughput, and regulatory targets and/or constraints.*

Software Tools for Process Design. Design optimization, as well as operation post commissioning, can benefit considerably from the use of modern (gradient-based) optimization algorithms and their software implementations [32, 78], yet the deployment of such optimization algorithms in commercial and practical applications is still relatively limited [31]. From a mathematical perspective, steady-state flowsheet simulation requires solving the material and energy balances of the process units, as well as the physical property cor-

relations or equation(s) of state for the chemical components present. The corresponding systems of algebraic equations (for steady-state process models) are often high-dimensional and typically highly nonlinear, ill-conditioned, and poorly structured [29].

Most commercial and industrial process flowsheet simulation tools use a sequential modular (SM) approach, simulating the interconnected unit operations (e.g., with dedicated unit operation solvers) in a process by "tearing" recycle streams and solving individual unit operations in sequence. The recycle streams are then converged in an interative manner by updating the simulation inputs until some convergence criterion is reached. While such SM approaches can solve process flowsheets starting from poor initial guesses, they are relegated to estimating gradients and sensitivities through finite difference calculations, which can be computationally expensive and potentially inaccurate [29, 189].

An alternative way to simulate and optimize process flowsheets, referred to as the equation oriented (EO) approach, is to solve all the model equations simultaneously [71]. The EO approach allows for custom unit operation models to be easily incorporated (since no separate solver is required) and simplifies process optimization by allowing for gradient matrices to be calculated via automatic or symbolic differentiation [71]. However, EO modeling environments generally rely on Newton-type solvers and require informed initial guesses to solve (and optimize) process flowsheet models successfully and reliably [230].

Several flowsheet simulation strategies use a "hybrid" approach to combine the benefits of EO process simulators in optimization with the robustness of SM process simulators [190]. These hybrid approaches typically are capable of calculating Jacobian and Hessian matrices via automatic differentiation and use SM simulation concepts to provide the informed initial guesses to solve process models. This step can require significant computational time and effort, particularly when the flowsheet is large and/or complex. Various numerical methods have been proposed to expedite the identification of good initial guesses and to provide

alternatives to the Newton class of methods, including: (i) homotopy continuation [163, 279], (ii) pseudo-transient continuation [190, 195], (iii) interval Newton methods [153, 225], and (iv) global-optimization-based methods [164], among others.

Survey Approach. A relatively recent survey [79] of industrial practitioners in systems and control reported that "optimization of a process or operation" ranks as the most important skill for chemical engineering graduates. As a result, my survey was based on the assumption that process design optimization tools are needed, implemented, and used—to varying extents—throughout the process lifecycle, including (i) research and development, (ii) engineering and construction, and (iii) execution and operation (Figure 2.1).

To investigate the validity of this assumption, my survey involved interviews with experts and practitioners working in the various stages of the process lifecycle. The interviews ranged from thirty minutes to an hour in length and were mostly conducted in-person, with a few over the telephone or video conference. The interviews were focused on participants from the United States and Canada, with a few interviewees from Europe. In total, 110 interviews were conducted. The roles of the employees interviewed ranged from plant production engineers to process design specialists and to senior management, and interviewees were selected from various industries (base chemicals/commodity, specialty and polymers, air separation, and oil refining). A breakdown of the interviewees by roles and industrial sectors is shown in Figure 2.2. The results in this chapter are distilled from these interviews, which are admittedly less quantitative than standardized polling mechanisms (e.g., surveys). Nevertheless, this approach allowed me to focus the interviews on the use of optimization techniques (in their various guises and interpretations) and obtain insights that a "one-size-fits-all" questionnaire may not fully reveal.

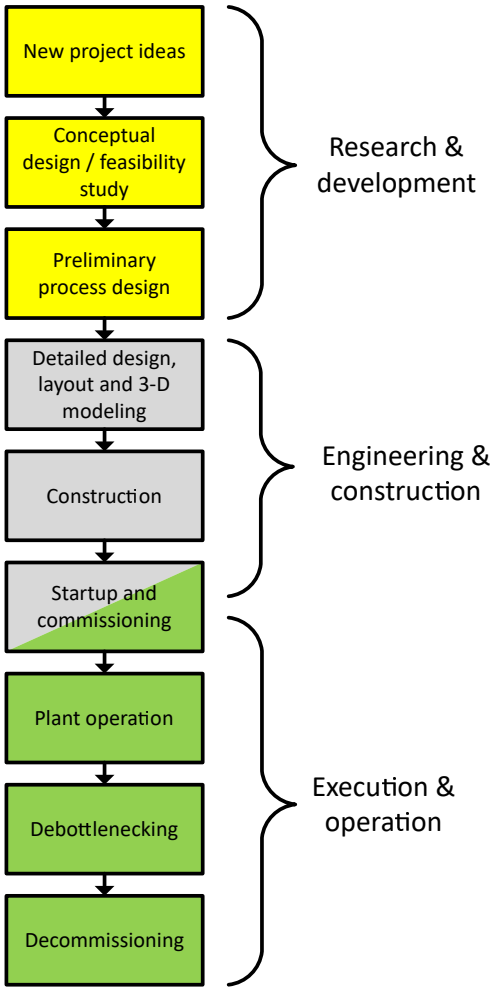

Figure 2.1: Typical lifecycle of a (petro)chemical manufacturing process.

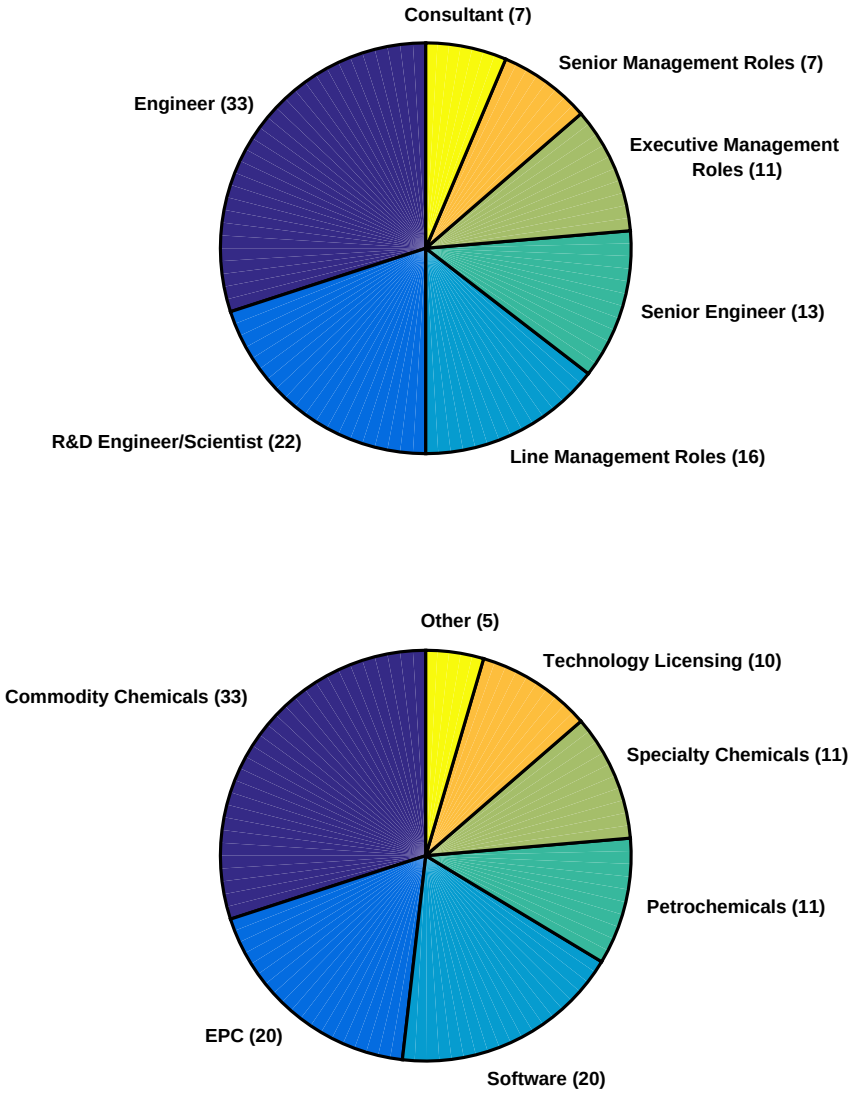

Figure 2.2: Structure of interviewee cohort by role in their respective organizations (top) and industrial sector (bottom).

2.3 Findings I: General Workflow for Process Design

The lifecycle of a (petro)chemical process (Figure 2.1) can be translated into functions and activities that belong to three main categories. Each functional category is typically the perogative of a specialized commercial entity:

- Technology licensing companies develop new process technologies (e.g,. a new catalyst), providing opportunities for the construction of new manufacturing plants or for capital improvements to existing plants. Such companies are also typically involved in the conceptual design of processes using their new technologies.

- Engineering, procurement, and construction (EPC) companies can aid in the conceptual design, and they specialize in the detailed engineering work and "putting metal on the ground," or physically installing the designed plant.

- Operating companies cover the capital investment and assume responsibility for plant operation, maintenance, and improvements until it is decommissioned.

The analysis presented in this chapter generally follows this division of labor between technology development/licensing companies, EPC firms, and operating companies; however, the ecosystem of the petrochemical and chemical industries is more complex. Although many companies predominately carry out one of these specialized roles, larger companies may carry out more than one of these activities in-house via separate, dedicated teams or divisions. The implementation of new technology ideas into physical production facilities is a capital-intensive process that is managed via complex and detailed contractual agreements and supporting organizational structures, particularly when the three functions mentioned above are performed by different corporate entities. With the segregation of work and expertise along the process lifecycle, the value added in each step differs for each entity in the ecosystem

and results in complex interactions, manifest, e.g., in the (lack of) exchange of information and intellectual property transfer.

2.3.1 Process Research and Development

Technology licensing companies derive most of their revenue from accumulating licensable patents/intellectual property and maintaining valuable trade secrets. My interviews revealed that the biggest challenge such technology developers and licensors face is that the petrochemical and chemical industries are historically risk-averse and slow to change. It is therefore difficult to convince a potential licensee company to be the first adopter of a new technology; many interviewees emphasized the existence of a "race to be second," whereby there is a distinct reluctance on the part of technology operators to be the first to adopt a new and potentially unproven technology. Conversely, the same operators were eager to rapidly deploy *before* their direct competitors new technologies that had been *already proven by another early adopter*. This pattern seems to be most prevalent in the refining industry, where low profit margins limit the amount of risk that corporate management is willing to take. Interviewees engaged in work further downstream, such as in specialty chemicals, revealed a slightly less risk-averse attitude.

In addition to the crucial role of technological maturity, the main factors involved in successfully licensing a technology are capital and operating costs (CAPEX and OPEX) against yield and product quality, motivating technology licensing companies to optimize total revenue. This effort must strike a balance between potential per-unit cost reductions and the risk associated with altering a process. Experimental data surrounding new technologies are often very limited, and pilot testing is costly and time-consuming. Consequently, many interviewees believed that the mathematical models used for process optimization are incapable of predicting process behavior when many variables are changed, and they therefore

focus on the largest contributors to throughput, CAPEX, and OPEX. Specifically, management and engineers in technology licensing companies often use a divide-and-conquer strategy, where "tornado" charts (Figure 2.3) serve to identify the most significant revenue and cost components of a new technology. On the basis of such charts, engineers utilize the most influential degrees of freedom to maximize profit. This activity is carried out either by manually testing various scenarios (spanning design options and operating cases) or by using a computational process optimization package.

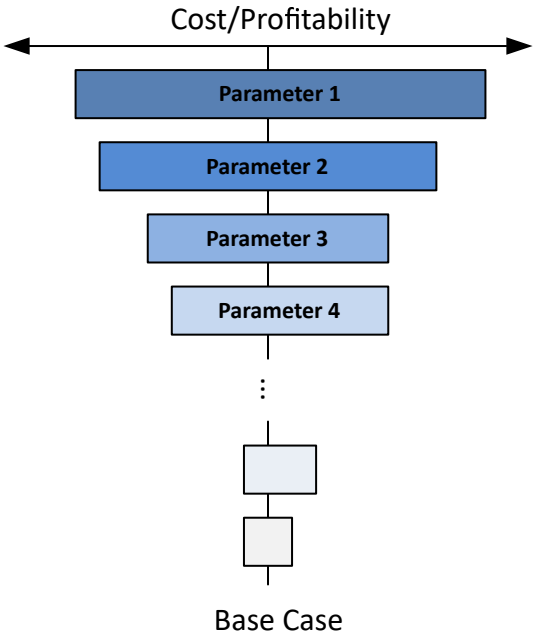

Figure 2.3: "Tornado" diagram for visualizing project cost sensitivities to various parameters.

The interviews also revealed information "silos" that prevent the use of process optimization tools, particularly by technology licensees. While often limited, pilot plant data can be (and are) used to build detailed mathematical models for at least parts of a process (e.g., the main reactor). These models are typically only used internally by the technology developer company and rarely shared with potential licensees in open form; rather, a "black box" model is provided along with the design-and-build blueprint. Operating companies

are thus often forced to create their own versions of the high-fidelity, "digital twin" models used internally by the technology licensor, employing dedicated equipment modeling teams to reconcile historical operating data. This state of affairs represents an impediment in the consistent application of optimization strategies at the unit and plant levels throughout the lifecycle of a process.

Interviewees from all three types of commercial entities emphasized that their companies have adapted and/or developed standardized workflows for evolving a new process technology from concept to construction. These workflows share many similarities to the Stage-Gate®[‡] approach [57, 125] for new product development and commercialization and involve a series of stages involving project work and data analysis, each followed by a "gate." Each such gate results in a go/no-go decision concerning the continuation of the project, and each subsequent stage is designed to gather further information and reduce key uncertainties about a technology. Later stages tend to be longer in duration and more capital intensive than earlier stages, which are dedicated to conceptual exploration. The earliest stage in this workflow is termed a conceptual design or feasibility study, based on a new business opportunity or laboratory-scale discovery. A validated concept then progresses through a series of front-end loading (FEL) stages (sometimes referred to as front-end engineering design, 'FEED,' or feasibility analysis), wherein designs are progressively refined and cost estimates are improved. The final stages for a new process typically involve detailed engineering—including full unit specifications, 3-D plant modeling, and relief calculations—and the ultimate decision on whether to construct the resulting designed plant.

Figure 2.4 shows a typical workflow for taking an idea for a new (petro)chemical process from engineering design to plant construction, as well as some of the typical levels

[‡]Stage-Gate® is a registered trademark of Stage-Gate Inc.

of uncertainty at various stages. In the vast majority of cases, the number of early-stage projects is considerably larger than those in the later/final stages of the pipeline. This attrition can be attributed to natural elimination of technically sound but economically suboptimal ideas (an empirical threshold of a 15% return on investment is often used), as well as to fluctuations in the type and number of business opportunities available at any given time.

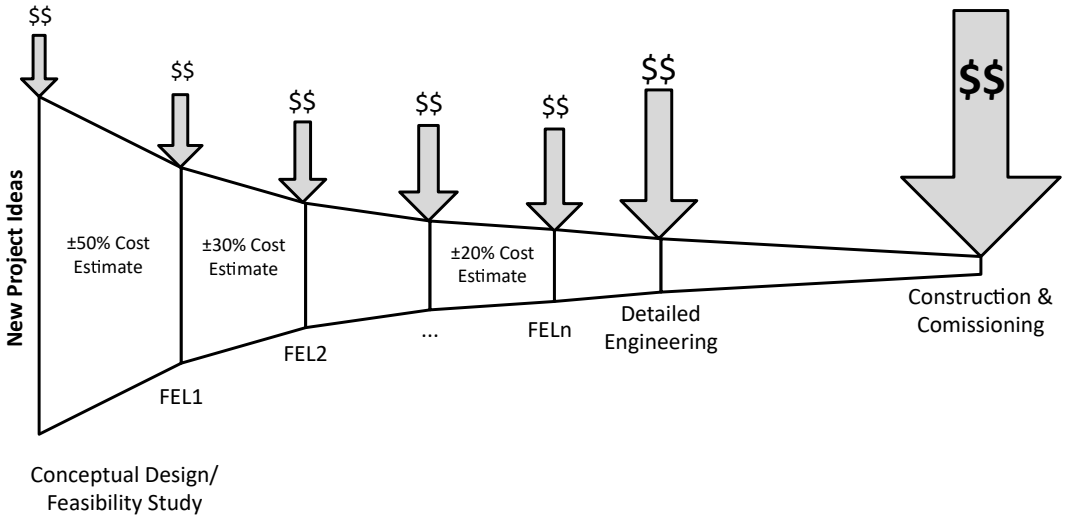

Figure 2.4: Workflow for process design engineering involving various stages. The arrows represent relative capital expenditures at each gate.

2.3.2 Detailed Engineering and Construction

Once a technology has withstood the scrutiny of the feasibility study (Figure 2.4), it is ready for detailed engineering design. This activity is typically undertaken by an EPC company, sometimes in close collaboration with the operating company that will own and operate the plant. The interviews revealed that the roles of the two entities and the distribution of labor depend on the respective levels of expertise and available manpower, ranging from complete design by the EPC to a side-by-side effort by design engineers from both the EPC

and operator. The main outcomes of this design effort are, (i) an accurate cost estimate for project execution, (ii) a complete design blueprint for procurement and construction, and (iii) a preliminary evaluation of potential project execution risks and issues. Given the amount of domain expertise required, detailed engineering design work is generally carried out by separate and specialized teams (focusing, e.g., on heat exchange, rotating equipment, piping and instrumentation, relief systems), with a well-defined central management, reporting, and interaction structure.

In the interest of accelerating the detailed design schedule and saving cost, many EPC company employees reported moving away from a "series" design workflow, in which each engineering team designs the section of the process they specialize in and passes it onto the next team, and towards a "parallel" approach, whereby the engineering teams must simultaneously complete their relevant tasks. The parallel workflow involves a block flow diagram (BFD) with heat and material balances specifying inputs and outputs for the engineering teams; the aforementioned reporting structure is used to notify of changes to these input/output values, triggered, e.g., by equipment limitations. Interviewees noted that this workflow makes changes to the process more costly as the detailed design process advances. Engineers tasked with carrying out detailed designs therefore focus on finding a solution that is feasible from the process BFD point of view, without necessarily being economically optimal (either locally at the individual block level or at the plant level).

In the particular case when only certain units or subsections of a larger process flowsheet (rather than a full plant design) are licensed from a third party, the lack of detailed model information for licensed technologies (owing to nonexistence or, as mentioned above, to the reluctance of licensors to share such models) further limits the ability of EPC and operating companies to explore a broad range of operating conditions at the detailed design and engineering stage.

EPC companies generate revenue from project execution and are keen on maintaining a robust project pipeline. Further, EPC firms are not directly vested or interested in the final operating stage of a new production facility. As a consequence, they historically tended to bill their clients on a reimbursable cost basis (charging hourly for services), ensuring that revenue is generated even when projects are stopped at an early stage/gate (Figure 2.4). On the other hand, the main customers of EPC firms, operating companies, are increasingly favoring lump-sum contracts, which are structured such that the EPC company guarantees a deliverable for a fixed sum and on a given schedule, assuming the financial risk if the cost exceeds the agreed-upon amount. Additional financial penalties can be imposed on schedule overruns and performance shortcomings on the delivered product. With the growing prevalence of lump-sum contract bids, EPC companies emphasize heavily the importance of accurate capital cost estimates, enabling place project bids low enough to win business, yet high enough to guarantee the project can be completed for the price.

As a consequence, EPC companies *very rarely perform any degree of process design optimization*, in the sense defined in Section 2.2. Rather, new process BFDs typically deviate very little from proven (and often quite old) designs, with the goal of maintaining feasibility and minimizing uncertainty in equipment design and cost estimates. This "pattern-based" approach by EPC companies is a natural response to the high risk and capital-intensive nature of the industry (and the associated risk-averse behavior of their customers), and potentially a significant contributor to the generally slow-changing nature of the petrochemical and chemical industries. EPC companies maintain a wealth of subject-matter expertise through historical records and senior experts, and they accomplish the important task of building functional, on-spec plants by relying on time-tested solutions.

2.3.3 Project Execution and Process Operations

Even though technology licensing companies and EPC companies may contractually guarantee performance for (parts of) a manufacturing process, the financial risk for a new (petro)chemical plant ultimately falls on the operating company that owns it. As they directly benefit from reduced operating costs or increased throughput, many large operating companies maintain in-house process R&D teams that engage in optimal process design at the conceptual design stage (to evaluate new capital investments) and/or in process debottlenecking. Personnel in such R&D teams were generally of the opinion that commercial tools for optimal process design suffer from usability and convergence issues; using them requires significant amounts of experience and training. Further, setting up optimization problems itself takes significant expertise (to identify constraints, feasible variable ranges, etc.), so a company must invest time and effort to built up the requisite knowledge for engaging in optimal process design. The biggest challenge reported by these expert process R&D teams was model validation. The engineers are often provided with incomplete data (or data of uncertain quality) from manufacturing plants and technology licensing companies' pilot plants, making *the development of predictive models for optimization a considerable challenge*.

At the front line of such operating companies are the manufacturing plants, which are staffed by plant managers, production engineers, and process operators. At the plant-management level, the primary concerns are ensuring regulatory compliance and safe operation, as incidents in these areas can result in a plant shutdown and lost production. Although not the primary concern, economic improvements are still important to plant managers, but they typically only have enough capital-improvement budget to invest in the "lowest-hanging fruit," or projects guaranteed to make the most return in the shortest time. Production engineers and process operators are tasked with keeping plants running smoothly, and thus spend most of their time dealing with operational issues. Although they may use computa-

tional simulation tools to identify potential issues in the plant, engineers at the plant level generally stated that *do not have the time to develop expertise in optimization tools*.

In the case of several operating companies, my survey revealed an additional barrier for optimal process design in the interaction between engineers at the plant level and the internal R&D teams. Plant personnel often decline to implement possible improvements identified by the internal R&D teams, based on factors that were not considered at the outset of the improvement initiative, including un-modeled plant behaviors such as the evolution of trace components, equipment degradation (e.g., increased fouling), or the advent of side reactions. Plant operators may also reject proposed process improvements because optimized processes are often highly integrated and/or operate very close to their bounds, making them inflexible and difficult to control. An operating point near plant equipment limits may also be more difficult to reach safely and quickly during start-up. Such outcomes can be attributed to poor project scope definition on the part of the R&D team.

The above limitations notwithstanding, my survey revealed that operating companies as a whole possess the most optimal process design expertise out of all three entities defined in Section 2.2. In addition to design optimization, many operating company R&D teams use the optimization capabilities of process simulation packages to facilitate the model validation process, identify debottlenecking projects, and define conceptual plant designs.

2.4 Findings II: Industry-Specific Insights

Although the value generation schemes for research and EPC firms do not vary much across different industries, operating companies in different industries face very disparate economic and technological challenges. This sections relates several industry-specific patterns identified through my survey.

2.4.1 Petroleum Refining

The petroleum refining industry converts the hydrocarbons in crude oil to transporation fuels, lubricants, and other products through catalytic reactions, thermal processes, separations, and blending processes [62]. The products typically have strict "macroscopic" quality specifications, such as octane number, flash properties, and/or sulfur content. Refineries may switch between different feedstock and product slates as often as daily, and ensuring consistent quality of the end-products via blending. With variable process inputs and outputs, *process design is not focused on optimality at a fixed steady-state operating point*. Rather, refineries are designed to allow for flexible operation with different grades and types of raw material. The design objective is to maximize product yield and plant throughput under such uncertainty; this deviates from the capabilities of standard steady-state process simulation and optimization packages (which offer deterministic optimization, rather than optimization under uncertainty and/or robust optimization).

In a different vein, engineers in operating companies in the refining industries reported the extensive use of computational optimization tools at the scheduling and planning layers, aiming to determine optimal feedstock purchase and production strategies for existing plants. Fluctuating petrochemical market prices make planning and scheduling vital to maximizing profits in the refining industry. The use of real-time optimization (RTO), a steady-state optimization of the targets/setpoints of the control system, carried out periodically (e.g., every hour), is very prevalent in the refining industry [175, 289]. These calculations typically involve solving a nonlinear program including pricing discontinuities, mass and energy balances, product properties, and separation thermodynamics to maximize profit or minimize cost, and practical implementation of RTO presents its own relavent challenges pertaining to imprecise economic data, model validation issues, and deviations from steady-state operation.

2.4.2 Commodities: Air Separations and Chemicals

Air separation and commodity/base chemical manufacturing plants operate with relatively low profit margins. Further, especially in the air separation case, they are treated as utility suppliers by their downstream customers, with explicit expectations related to product quality and availability. Unlike the refining case above, the feedstock and product slates are relatively constant in time, allowing for *significant usage of optimal process design tools to maximize competitiveness*. Operating companies in these industries maintain teams of in-house process R&D experts and have accumulated significant process know-how, which is captured, among others, in accurate process models.

Many internal process R&D teams that I interviewed at air separation and commodity chemical companies reported using equation-oriented optimization tools, with technical challenges mainly arising from difficulty of use. A special obstacle is posed by the fact that no single software tool offers all the desired features, such as equation-oriented modeling, global optimization, and simulating process start-up and failure events. Engineers are thus forced to migrate models and/or significant amounts of data between different software tools. Plant models are thus often created in multiple software packages and must be carefully cross-validated, a time-consuming and potentially error-prone process. Another reported challenge was related to presenting the optimization results and clearly defining the associated benefits to stakeholders.

With significant intellectual property and effort invested in commercial software tools, operating companies are often reluctant to switch software packages, but many still continually monitor new software releases to ensure they are at the forefront of technology. The required financial investment and the difficulty of using commercial optimization packages is also prohibitive to smaller companies seeking to adopt and deploy optimization techniques.

While larger companies may be willing to invest in adopting (or at least testing) new software tools, allowing them to evaluate the economic benefit directly, the limited budgets of smaller companies force them to only purchase software that is deemed essential, and (expensive) design optimization packages often to not pass this criterion.

2.4.3 Specialty Chemicals

Companies involved in specialty chemicals production often develop new products that can be made from a limited array of (purchased) feedstock. Here, margins are high (compared to the commodities sector), and speed-to-market is of utmost importance [231]. To protect intellectual property surrounding new products, many specialty chemical companies perform the research and pilot plant functions themselves, only seeking external assistance when necessary, such as in detailed plant design. The combination of limited knowledge about product chemistry, smaller manufacturing plant sizes, and higher profit margins leads *optimal process design to be a lower priority for most specialty chemical companies.*

To minimize time-to-market for new products, plants may be designed after only a basic exploration of the reaction chemistry or physical properties of the new compounds. Process engineers are thus facing the challenge of producing a feasible (rather than optimal) design, and the development teams often spend the bulk of their time validating physical properties for proprietary chemical components. In addition, lower production rates motivate many companies to opt for batch processes.Interviewees reported that, if ever desired, scaling up of a specialty chemical production process poses a significant challenge due to the difficulty in predicting process behavior with a limited understanding of the chemistry and physics.

The relatively high profit margins in the specialty chemicals sector render minimization of energy use in manufacturing a secondary concern. Additionally, the lack of full system understanding limits the potential impact of optimization tools, and processes are

generally "optimized" by physical experimentation—changing operating points in the actual process and observing the effects. The major goals in such optimization routines for a production process in this industry are thus to maximize the production rate/throughput (often while only using existing equipment), minimizing waste generation, and meeting regulatory concerns.

2.5 Findings III: Overarching Trends

A recurring theme encountered in my interviews was the difficulty in demonstrating the value of investing in optimal process design capabilities. For established processes, it is difficult to estimate the extent to which a process can be improved beyond the experience-guided, trial-and-error operational changes that were often made over the span of decades. Likewise, for new process concepts, it may be unclear how much a process can be improved based on models constructed from the limited information and experimental data available.

As a related organizational challenge, it became evident that engineers who could benefit from using optimization software seldom have the latitude to select, purchase and adopt new software tools to supplement or replace existing, established ones. This is especially true in larger companies, where the user base for incumbent tools could be quite large, and retraining this workforce would be a massive undertaking. Process R&D teams must get management approval to purchase software tools or training, but—as revealed before—such initiatives are undermined by the difficulty of quantifying the expected benefit of computer-aided optimization and calculate returns on investing in new software.

A second recurring theme in the survey was misalignment in incentives and constraints among technology licensing, EPC, and operating companies. Figure 2.5 presents a summary of the objectives and constraints encountered by each of these entities, as well as the flow

of information between them. Note that each entity has its own objective, and information and knowledge are compartmentalized, leading to a disconnect between those who possess detailed, predictive models (if they exist) and those who would directly financially benefit from process optimization. It was apparent through my interviews that engineers at many companies felt they were building models or collecting data that a different company already had, but would not share for business reasons.

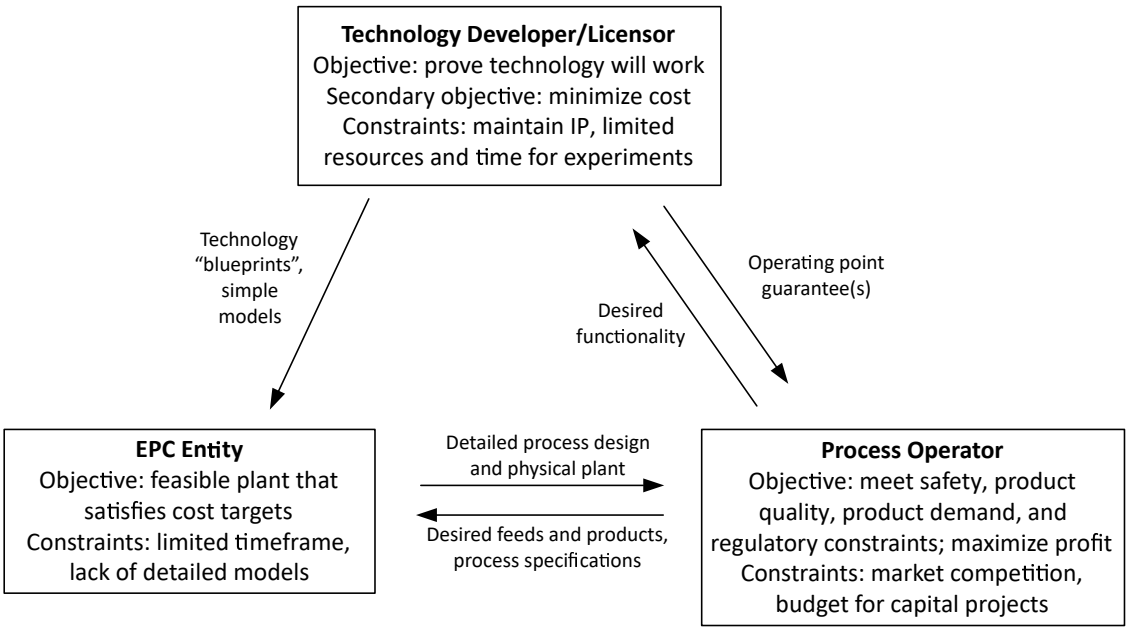

Figure 2.5: General objectives and constraints of entities engaged in the process design ecosystem.

The final recurring theme from the interviews was that, aside from dedicated R&D scientists, employees in the petrchemical and chemical industries are cautious and hesitant to change. Owing to the magnitude of capital investments in the industry and the repercussions of safety and/or regulatory violations, the primary concern of practitioners at all levels is to ensure that processes operate reliably and predictably. There is considerable uncertainty in the design of new processes, and new operating schemes introduced by a deterministic

optimization procedure—whose inner workings and decisions may not be fully transparent to the user—are often met with skepticism by plant operators, who may have an empirical understanding of the plant that goes beyond the information captured in the optimization model.

2.6 Perspective

Although the potential economic improvements of large-scale process optimization can be considerable, this survey of industrial experts and practioners revealed a limited penetration of such techniques into industry workflow and applications. The multitude of practical challenges faced by process engineers in their day-to-day responsibilities often preclude them from developing the requisite expertise or the predictive models involved in optimal process design.

These interactions with a broad cadre of industry experts suggest that increasing the adoption of process optimization tools in practice rests on three pillars:

(i) **Accessibility:** ensuring the future *seamless integration* of optimization capabilities and custom, detailed modeling in the process simulation software tools that are already familiar to industry practitioners. The implementation of optimization capabilities should target all aspects of usability: an easy to use interface for problem definition, transparent troubleshooting, fast computation and a detailed presentation and interpretation of the results. At the modeling level, the ability to deal with complex, physical models, as well as with subject matter expert knowledge (described in the form of, e.g., spreadsheets) would be very valuable. Furthermore, it is necessary to incorporate capabilities for model validation and data reconciliation, allowing engineers to minimize model uncertainty.

(ii) **Alignment and information availability:** sharing optimization-relevant information between the entities involved in the process design and operations ecosystem. In particular, starting at the equipment manufacturer and technology development level, each process or process concept could be accompanied by a "digital twin" model that can be used further downstream by EPCs and operators.

(iii) **Awareness and training:** the undergraduate chemical engineering curriculum provides limited exposure to optimization concepts [79], and graduate-level training is similarly limited[80]. As such, many engineers are not fully aware of the opportunities afforded by process optimization. Mandating that such concepts be taught at the undergraduate level is unrealistic given an already very full curriculum. A potential avenue is the expansion of corporate training programs.

The results of this survey of current practices and challenges in optimal process design strongly motivated the research directions taken in this dissertation.

Part I

Multi-Resolution Process Design Optimization

Chapter 3

Modeling, Simulation, and Optimization of Process Flowsheets using Pseudo-Transient Continuation: Preliminaries[†]

3.1 Overview

As detailed in Section 2.2 of Chapter 2, "hybrid" strategies for solving process flowsheets seek to combine the optimization-related advantages of equation-oriented modeling environments with the robustness of sequential-modular packages. In this chapter, I review pseudo-transient continuation as a promising technique to generate informed initial guesses for process flowsheet simulation and optimization. The pseudo-transient framework has proven to improve the robustness of process flowsheet simulation and optimization [190], addressing several usability challenges in Chapter 2. In view of the challenges described in Section 2.6, Part I this dissertation examines how pseudo-transient concepts can extend to *complex, physical models, as well as with subject matter expert knowledge.*

This chapter reviews several important concepts and mathematical tools involved in pseudo-transient flowsheet modeling, simulation, and optimization.

[†]The contents of this chapter are largely based on the following two publications (C.T. contributed to the analysis and writing of the first manuscript and is the primary author of the second):
R.C. Pattison, C. Tsay, and M. Baldea. Pseudo-transient models for multiscale, multiresolution simulation and optimization of intensified reaction/separation/recycle processes: Framework and a dimethyl ether production case study. Comput. Chem. Eng., 107:161–172, 2017.
C. Tsay and M. Baldea. Fast and efficient chemical process flowsheet simulation by pseudo-transient continuation on inertial manifolds. Comput. Method. Appl. M., 348:935–953, 2019.

3.2 Process Flowsheet Simulation and Optimization

Mathematically, the simulation of a process flowsheet involves solving the equations describing the material and energy balances of the involved unit operations, as well as correlations defining the relevant physical properties, reaction rates, and phase equilibria. Most chemical processes are integrated and contain recycle streams (Figure 3.1, left), which introduce algebraic loops [15].

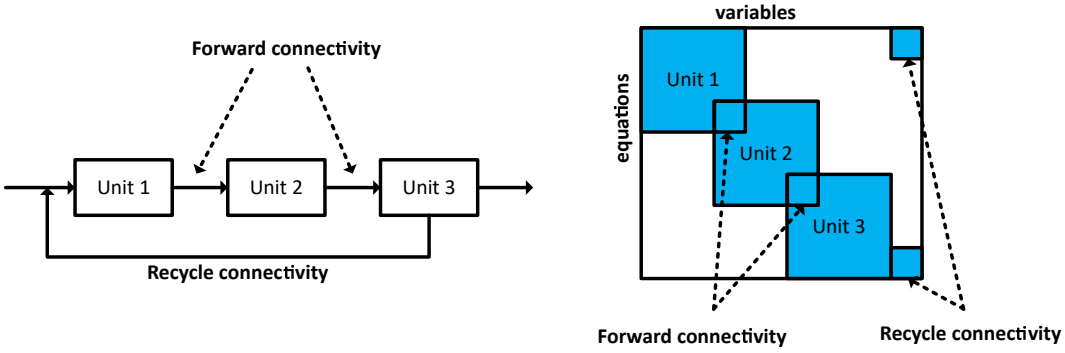

Figure 3.1: Problem structure for a flowsheet with three units and a recycle stream.

The corresponding system of equations is typically non-linear, large-scale, and poorly conditioned, and Newton-type solvers require an initial guess that is "close" to the solution in order to converge [163]. In general, the flowsheet constitutes a system of n algebraic equations:

$$\mathbf{f}_{ss}(\mathbf{x}, \mathbf{z}, \boldsymbol{\theta}) = \mathbf{0} \tag{3.1}$$

with $\mathbf{f}_{ss} : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$. A subset of the variables in the process flowsheet model are *decision variables*, $\mathbf{z} \in \mathbb{R}^m$, that can be selected or optimized, and the remaining variables, $\mathbf{x} \in \mathbb{R}^n$, are calculated by solving the model equations, $\mathbf{f}_{ss}(\cdot) = \mathbf{0}$. $\boldsymbol{\theta}$ includes parameters that are not free to be optimized. As shown in Figure 3.1 (right), the unit-based representation of a process flowsheet naturally gives rise to a "block" structure within the model equations. Moreover, the system is nearly block-banded, with some off-band elements introduced by

the connectivity equations of recycle streams that correspond to algebraic loops. Note that the system has more variables than equations; the variables in $\mathbf{z}$ and $\boldsymbol{\theta}$ must be defined (set) to fully specify and simulate the system.

Finding a solution for such systems, an effort referred to as "flowsheet simulation" or "process simulation," is quite challenging. Newton-type methods are only locally convergent for nonlinear systems, but are highly efficient and converge to the solution quickly if an initial guess within the "convergence basin" is provided. The two main challenges limiting process flowsheet simulation with Newton-type solvers are initialization and scale [299]. Since Newton-type methods require computing a Jacobian matrix at each iteration, computational cost increases significantly with model size. However, as algorithmic and hardware-related parallel computing advances continue to address the scaling challenges of Newton-type methods, identifying good initial guesses for initialization becomes the predominant concern, and the definition of rigorous procedures for obtaining such initial solutions remains an open problem. As a consequence, the solution of the entire flowsheet model is rarely pursued (or possible) in practice, and current computational approaches predominantly rely on informed decompositions to reach a solution [32, 256]. For example, early decomposition strategies for process flowsheet simulation exploited their physical structure [240, 282], as suggested in Figure 3.1, partitioning process flowsheet models into "blocks" (typically corresponding to individual unit operations), that are solved separately.

3.3 Pseudo-Transient Continuation

Pseudo-transient continuation (PTC), or pseudo-time stepping, is a numerical technique developed for solving complex problems described by ill-conditioned algebraic models. The method has been discussed in the literature primarily in the context of solving discretized partial differential equations (PDEs), such as Monge-Ampère type equations [10],

Poisson-Bolzmann equations [233], and Navier-Stokes type equations [51]. However, Pattison and Baldea [190] recently demonstrated the relevance of PTC as a technique for process flowsheet simulation and optimization (see also [192, 193]), and several studies [59, 159, 160] later demonstrated its application to complex process optimization.

The method relies on converting an algebraic system of the form (3.1) to a system of *differential-algebraic equations*, or DAEs. This conversion can be performed, among other strategies, by considering the natural dynamics of the process. The solution to the original system (3.1) is found by (i) providing initial conditions for the differential variables of the DAE system, and (ii) carrying out (pseudo-)time integration until steady state is reached. Below, I briefly review several concepts regarding DAE systems and their solution. Further details can be found in the books by Brenan et al. [37] and Cellier and Kofman [49].

Differential-Algebraic Equations. DAE systems take the general form:

$$\dot{\mathbf{a}}(\hat{t}) = \mathbf{f}'(\mathbf{a}(\hat{t}), \mathbf{b}(\hat{t}), \boldsymbol{\theta}, \hat{t}) \tag{3.2}$$

$$\mathbf{0} = \mathbf{g}'(\mathbf{a}(\hat{t}), \mathbf{b}(\hat{t}), \boldsymbol{\theta}, \hat{t}) \tag{3.3}$$

$$\mathbf{a}(0) = \mathbf{a}_0 \tag{3.4}$$

where $\mathbf{a}$ denote the dynamic (state) variables, $\mathbf{b}$ denote the algebraic variables, and $\hat{t}$ is the pseudo-time variable. Process system models are typically described by index-1 DAEs, where the *index* of a DAE system (3.2)–(3.4) is defined as the minimum number of differentiations that must be performed on the algebraic equations $\mathbf{g}'$ in order to obtain ordinary differential equations for the algebraic variables $\mathbf{b}$. For example, the dynamic variables $\mathbf{a}$ may comprise temperatures and compositions from (dynamic) mass and energy balance equations, while $\mathbf{b}$ include properties computed from equations of state or other constitutive relations.

The simulation of DAE systems comprises two steps:

- *Consistent initialization* refers to solution of the algebraic equations $\mathbf{g}'$ to find $\mathbf{b}(0)$ that is consistent with the initial conditions $\mathbf{a}(0) = \mathbf{a}_0$. Typically this step is performed using an iterative nonlinear solver. The algebraic equations of index-1 DAE systems exhibit the property that $\partial \mathbf{g}'/\partial \mathbf{b}$ is nonsingular.

- *Time integration* refers to simulating the system using a time-stepping routine. This is typically performed using implicit methods (e.g., backwards difference schemes) that require the solution of a nonlinear algebraic equation system at each time step. Variable-length time steps can balance computational efficiency and numerical accuracy.

Convergence of PTC. Kelley and Keyes [128] showed that, as the size of the time step taken during pseudo-time integration approaches infinity, the PTC method is equivalent to applying Newton's method to the original algebraic system. Taking smaller time steps in PTC effectively uses previous time steps to provide good initial guesses for the ensuing time step (e.g., if a Newton-type solver is used during implicit time integration). Since the end goal of PTC is to recover a steady-state solution, the time integration is not physically meaningful—it is purely a tool for numerical convergence. The convergence properties of PTC are further discussed in Kelley and Keyes [128] for ordinary differential equations (ODEs) and in Coffey et al. [55] for DAE systems.

Remark 3.1. There are several key differences between PTC and homotopy continuation (HC), which has also been applied in the context of process flowsheets [21, 117]. For instance, PTC recovers the problem solution at the steady state of the dynamic model (i.e., the time derivatives of state variables are equal to zero), while the solution in HC is not a steady state of the dynamic system. Moreover, the state variable dynamics in HC are dependent

on the inverse of the Jacobian, which can be an issue if the Jacobian becomes singular at any point during time integration [117]. On the other hand, the state variable dynamics in PTC are dependent on the original algebraic functions $\mathbf{f}_{ss}$. A more complete discussion of the differences between PTC and HC, as well as the comparative advantages of PTC, can be found in the Appendix to Pattison and Baldea [190].

3.3.1 Application to Process Flowsheets

PTC is applied to a process flowsheet by converting a subset of the equations in $\mathbf{f}_{ss}$ to ODEs, converting (3.1) into a DAE system of the form:

$$\mathbf{f}(\boldsymbol{\tau} \circ \dot{\mathbf{x}}_d, \mathbf{x}_d, \mathbf{x}_s, \mathbf{z}, \boldsymbol{\theta}) = \mathbf{0} \tag{3.5}$$

$$\mathbf{g}(\mathbf{x}_d, \mathbf{x}_s, \mathbf{z}, \boldsymbol{\theta}) = \mathbf{0} \tag{3.6}$$

where $\mathbf{f} : \mathbb{R}^{2n_d} \times \mathbb{R}^{n_s} \times \mathbb{R}^m \rightarrow \mathbb{R}^{n_d}$ and $\mathbf{g} : \mathbb{R}^{n_d} \times \mathbb{R}^{n_s} \times \mathbb{R}^m \rightarrow \mathbb{R}^{n_s}$ are the set of differential, and algebraic equations, respectively, and $\mathbf{x}_d$ and $\mathbf{x}_s = \mathbf{x} \setminus \mathbf{x}_d$ are the differential and algebraic variables. A vector of user-defined time constants $\boldsymbol{\tau}$ defines the rate of change of each pseudo-transient variable ($\boldsymbol{\tau} \circ \dot{\mathbf{x}}_d$ denotes the Hadamard product of $\boldsymbol{\tau}$ and $\dot{\mathbf{x}}_d$). Per the above discussion of PTC, there are three goals that must be met when selecting the set of dynamic variables and reformulating their corresponding equations into ODEs: (i) the consistent initialization step must solve reliably, (ii) the dynamics of the resulting DAE model must be stable, and (iii) the steady-state solution of the pseudo-transient model must be exactly the same as that of the original algebraic process model. Each of these key goals is further described below:

(i) **Consistent initialization**: the algebraic variables $\mathbf{x}_s$ must be computed at $\hat{t} = 0$ by solving solving $\mathbf{g}(\mathbf{x}_d^0, \mathbf{x}_s, \mathbf{z}, \boldsymbol{\theta}) = \mathbf{0}$ for $\mathbf{x}_s^0$. This can be accomplished, for example, by

selecting $\mathbf{x}_s \subset \mathbf{x}$ such that consistent initialization involves only solving a full rank linear system of equations:

$$\mathbf{g}(\mathbf{x}_d^0, \mathbf{x}_s, \mathbf{z}, \boldsymbol{\theta}) = \mathbf{A}(\mathbf{x}_d^0, \mathbf{z}, \boldsymbol{\theta})\mathbf{x}_s + \mathbf{B}(\mathbf{x}_d^0, \mathbf{z}, \boldsymbol{\theta}) \quad = \quad \mathbf{0} \tag{3.7}$$

(ii) **Pseudo-time integration**: the pseudo-transient DAE system is normally solved using a variable step DAE integrator. While stability is not trivial to establish for a general nonlinear dynamical system, various stability properties are provided in the pseudo-transient model libraries developed in previous works [190, 192, 193], where it was shown that the numerical integration of the pseudo-transient DAE systems reliably converges to the steady-state solution. Dynamic stability is generally achieved by proper selection of the time constant magnitudes in $\boldsymbol{\tau}$. In turn, the selection of time constants is based on physical arguments; equations are dynamically decoupled via a hierarchy of time scales present in process flowsheets, as shown in Figure 3.2. Thus the application of PTC to process flowsheets can be viewed as a *dynamic decomposition* of the model equations.

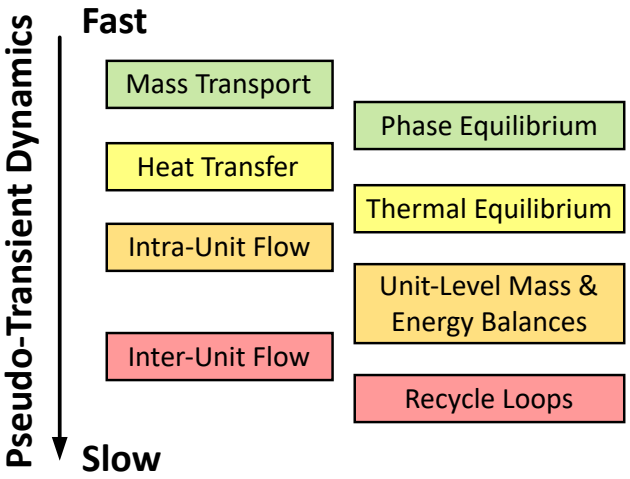

Figure 3.2: Pseudo-transient dynamics are defined to reflect a natural hierarchy of time scales in the phenomena that influence process dynamics.

(iii) **Static equivalence:** the pseudo-transient process modeling framework relies on the fact that the steady state solution of the reformulated pseudo-transient model is the same as that of the original algebraic process model, *i.e.,* when the time derivatives $\dot{\mathbf{x}}$ are zero, the values of the variables $\mathbf{x}_d$ and $\mathbf{x}_s$ are equivalent to those of their counterparts, $\mathbf{x}$, at the solution to the original algebraic model (3.1). I define the notion of static equivalence more precisely below.

Definition A dynamic system $\mathbf{f}(\dot{\mathbf{x}}, \mathbf{x})$ is *statically equivalent* to an algebraic system $\mathbf{f}_{ss}(\mathbf{x})$ if a steady-state solution of $\mathbf{f}$, defined as $\mathbf{x}^*$, s.t. $\mathbf{f}(\dot{\mathbf{x}} = \mathbf{0}, \mathbf{x} = \mathbf{x}^*) = \mathbf{0}$ is also a steady-state solution of $\mathbf{f}_{ss}$, s.t. $\mathbf{f}_{ss}(\mathbf{x} = \mathbf{x}^*) = \mathbf{0}$ [190].

Pattison and Baldea [190] created a library of pseudo-transient models for common unit operations (reactors, compressors, flash tanks, distillation columns, etc.) that satisfy goals (i)–(iii). In practice, time constants spanning several orders of magnitude tend to give good simulation results [190]. The resulting system is in a standard (multiply) singularly perturbed form [137]. Once stability is established for a time scale, the corresponding dynamics converge to and remain at a quasi-steady state, and the system evolution is governed by dynamics in slower time scales. PTC has been shown to expand the basin of initial guesses for solving process flowsheets, enabling the simulation and optimization of large flowsheets and complex unit operations without the close initial guesses often required by Newton-type solvers.

Flowsheet optimization is performed through a time relaxation-based algorithm [190], derived from the sequential methods for dynamic optimization of DAE systems [268]. Given a set of initial conditions for the differential variables of the DAE system and initial guesses for the optimization decision variables, the model equations are solved at each optimization iteration by integrating the DAE system to steady state (without including path constraints)

and then calculating the objective function, inequality constraints, and their respective gradients (based on expressions derived through symbolic or automatic differentiation). This information is used by gradient-based optimization solvers to update the decision variables, and the process is repeated until an optimization criterion is met. Notably, the equality constraints are solved at each iteration via pseudo-transient time integration, representing a "feasible path" approach, where optimization is performed only in the reduced space of the decision variables.

3.3.2 Reformulation of Process Flowsheet Equations

PTC can be applied naturally to the equations of process flowsheets by (i) modulating first-principles dynamic equations with a time constant in $\boldsymbol{\tau}$ or (ii) introducing first-order filters for equality constraints. However, several other strategies are employed in applying PTC to process flowsheets, which are described fully in [190] and summarized below:

Definition of temperature as a state variable: Typically, both the material and energy balance equations are nonlinear functions of temperature, leading to difficulties at the consistent initialization step. To address this, temperature is defined as a state variable, converting the steady state energy balance (3.8) to an ODE (3.9), where the derivative of temperature in a unit operation with respect to pseudo-time is proportional to the residual of the static energy balance equation:

$$0 = H_{in} - H_{out} + H_{gen} + Q_{ext} \tag{3.8}$$

$$\tau_e \frac{H_{in}}{T_0} \frac{dT}{d\hat{t}} = H_{in} - H_{out} + H_{gen} + Q_{ext} \tag{3.9}$$

$$T(\hat{t} = 0) = T_0 \tag{3.10}$$

where H denotes total enthalpy, T is the temperature, T_0 is the initial condition (initial guess) for the temperature, and τ_e is an arbitrary time constant.

Modulation of source and sink terms: Source and sink terms in material and energy balance equations may contribute significantly to stiffness and nonlinearity of the overall model. A state variable $\hat{\alpha}$ is introduced to modulate source/sink contributions, eliminating them at the initialization step (i.e. $\hat{t} = 0$) and fully representing them at the steady-state solution. A parameter $\hat{\alpha}$ is typically defined for each unit with source/sink terms that multiplies those terms. $\hat{\alpha}$ is initially zero and is formulated to increase to unity at steady-state.

$$\begin{aligned} \tau_\alpha \frac{d\hat{\alpha}}{d\hat{t}} &= 1 - \hat{\alpha} \\ \hat{\alpha}(\hat{t} = 0) &= 0 \end{aligned} \tag{3.11}$$

Dynamic tearing of recycle streams: A "tearing-like" approach is employed to treat recycle streams, whereby the upstream state $\mathbf{X}_{tear}$ of each recycle stream is given initial conditions to compute the inlet and outlet states of downstream units during consistent initialization. A first-order filter is applied to decouple the upstream states $\mathbf{X}_{tear}$ and downstream states $\mathbf{X}$, such that $\mathbf{X}_{tear} = \mathbf{X}$ at steady-state of the DAE system.

$$\tau_r \frac{d\mathbf{X}_{tear}}{d\hat{t}} = \mathbf{X} - \mathbf{X}_{tear} \tag{3.12}$$

The time constants, τ_e, τ_α, τ_r, (specifically, their relative values) are tunable in order to achieve desired stability and/or convergence properties.

Chapter 4

Multiscale, Multi-Resolution Modeling and Optimization of Intensified Reaction/Separation/Recycle Processes[†]

In this chapter, I examine the modeling, simulation, and optimization of process flowsheets incorporating multiscale physical models using pseudo-transient continuation. Specifically, this chapter shows how detailed, multiscale models can be easily embedded in a pseudo-transient reaction/separation/recycle process flowsheet model which also includes rigorous (although not necessarily multiscale) models of the separation systems and recycle streams, thereby creating a *multi-resolution flowsheet model*. I apply these new concepts to the design optimization of an integrated/intensified dimethyl ether production process featuring a quenched, adiabatic, packed-bed catalytic reactor and a dividing-wall distillation column. The case study shows that the optimal design of the dividing wall column is highly dependent on the level of detail included in the reactor model, amply justifying the need to utilize detailed, multi-scale models of unit operations and process equipment for the design of integrated/intensified processes. The presentation in this chapter follows closely the material published in Pattison, Tsay, and Baldea (2017) [197].

[†]The contents of this chapter are largely based on the following publication: R.C. Pattison, C. Tsay, and M. Baldea. Pseudo-transient models for multiscale, multiresolution simulation and optimization of intensified reaction/separation/recycle processes: Framework and a dimethyl ether production case study. Comput. Chem. Eng., 107:161–172, 2017. C.T. contributed to the analysis and writing of the manuscript.

4.1 Modeling of Intensified or Unconventional Process Units

The results presented in Chapter 2 revealed that global competition and regulatory constraints are driving the chemical and petrochemical industries to improve the economic and environmental performance of their processes. Specifically, manufacturers strive to maximize the production of product per unit of operating expenses (i.e. raw materials and utilities) while simultaneously minimizing capital expenses and meeting strict demand, product quality, safety, environmental, and process operating constraints. *Process intensification*, defined as the unconventional design and/or operation of processes that results in substantial performance improvements, represents a promising route towards these goals.

To optimize the design of a process including *intensified* equipment, engineers may therefore require the flexibility to represent "unconventional" or non-standard process equipment models at the flowsheet level. Note that Chapter 2 also highlights the importance of including non-standard unit models in flowsheet simulation and optimization. For example, consider the reactor unit operations available in many standard flowsheet software packages, which include stoichiometric reactors, equilibrium reactors, stirred tank reactors, and (simplified) plug flow reactors. There is no immediate way to incorporate (and accurately and completely describe) intensified, mulfunctional designs, such as membrane reactors or heat exchanger reactors. Similarly, intensified separation equipment such as dividing-wall columns must be captured using ensembles of interconnected distillation columns.

Furthermore, accuracy in predicting the performance of such intensified equipment is crucial for ensuring adoption by industry, especially considering that operating ranges may be diminished compared to conventional setups [182]. The accuracy of the models for intensified equipment can be improved by capturing the multiscale phenomena occurring in the aforementioned devices, such as diffusion into catalyst pellets, detailed reaction kinetics,

adsorption, membrane diffusion, heat transfer, flow distribution. Increasing the level of detail used in describing such equipment in flowsheet models allows for all of the relevant design decisions (e.g., reactor volumes, catalyst pellet sizes, cooling jacket volumes) to be considered simultaneously, such that the process design is optimized in a holistic manner.

While the need for such multiscale flowsheet modeling, simulation and optimization is clear, the literature reporting the inclusion of detailed, first-principles models of process equipment in process flowsheet modeling and optimization is limited. Existing efforts fall into two main categories:

1. *Surrogate modeling*, whereby simplified models are developed from scenario data generated by simulations of a more complex model, and *reduced-order modeling*, whereby low-order models are derived from full-order, high dimensional models using model reduction techniques (e.g., singular perturbation arguments [15]). For example, Dowling et al. [72] report the optimization of an oxy-combustion process with surrogate boiler model. A reduced-order 1D/3D boiler model and a trust-region method were used to carry out the derivative-free optimization calculations for the boiler simultaneously with the equation-oriented process flowsheet. Other works in the oxy-combustion literature have attempted similar flowsheet optimization approaches using surrogate models for the boiler [81, 82, 88]. Several other works employed surrogate models to approximate the input-output behavior of detailed unit models and embed these in process flowsheets, including [42] where a rigorous non-isothermal plug flow reactor model is used, and [247] where a detailed rate-based distillation column is used.

   The evident advantages of surrogate modeling are the reduction in problem size, as well as the option of using model functional forms that lend themselves more easily to use in optimization calculations. Moreover, surrogate modeling generation can be coupled

with a tailored optimization procedure [35]. On the other hand, the validity of surrogate models is confined to the domain spanned by the data used in their construction, and extrapolation presents obvious risks. These issues are to some extent avoided by using reduced-order models derived from a physics-based full-order system representation. Nevertheless, in many cases, the pathway for deriving such models is not obvious, and its implementation may require significant domain expertise.

2. A small number of literature works report the use of *rigorous, first-principles, and equation-oriented models*, predominantly for chemical reactors, in the flowsheet design optimization problem directly. These include the papers by Rodriguez et al. [212], where a detailed multi-tubular reactor model is included in the optimal design of a propylene oxide process, and Recker et al. [208], where a detailed multi-tubular differential sidestream reactor model is included in the optimal design of an ethyl tert-buthyl ether process. The use of such models is clearly desirable, but comes at the cost of high model dimensionality as well as model stiffness/ill conditioning, factors that create significant challenges for numerical simulation and optimization.

The literature overview provided above suggests that incorporating detailed physical models of process equipment (including intensified subsystems) into a process flowsheet model (potentially featuring numerous other "conventional" unit operations) is a significant challenge [54, 208]. These challenges may be related to difficulties in solving the process flowsheets simultaneously with detailed equipment models, especially in equation-oriented strategies, as reviewed in Chapter 3. Recall that, while they have significant benefits in optimization, equation-oriented flowsheet strategies require a method for solving a large system of nonlinear (and potentially discontinuous) algebraic equations. In this chapter I show how the pseudo-transient concepts reviewed in Chapter 3 can be applied to the modeling and

optimization of multiphase reactor units with an arbitrary degree of complexity. I then apply these new concepts to the design optimization of an integrated and *intensified* dimethyl ether production process to demonstrate the need in practice to utilize detailed models of unit operations for the design of integrated/intensified processes.

4.2 Multi-Resolution Pseudo-Transient Process Models

In this section, I concentrate on the reaction/separation/recycle prototype process structure shown in Figure 4.1, and I establish a framework for including detailed process equipment models into a pseudo-transient, multiresolution process flowsheet representation. My focus is on, (i) reactive systems (e.g., tubular reactors) with models featuring multi-physics phenomena that are spatially distributed over multiple length scales, and, (ii) vapor-liquid equilibrium based separation systems. Nevertheless, the framework is general and can be applied to the models of other process entities with similar structure and features.

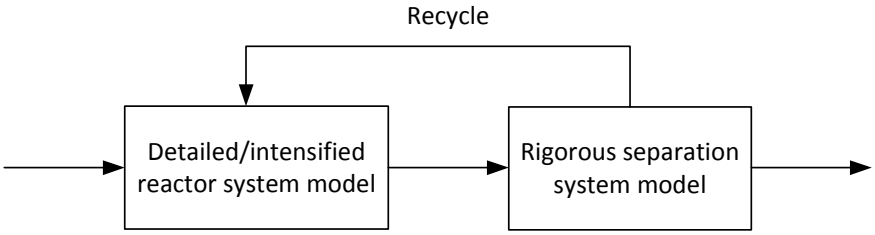

Figure 4.1: Simplified depiction of a process flowsheet including a detailed or intensified reactor model, a rigorous separation system model, and a recycle stream.

4.2.1 Multiscale Pseudo-Transient Reactor Models

The governing equations for the material and energy balances in the fluid phase of a non-isothermal distributed parameter reactor model are given most generally by the

convection-diffusion-reaction equation:

$$\frac{\partial u}{\partial t} = \nabla \cdot (D \nabla u) - \nabla \cdot (vu) + R \tag{4.1}$$

where u is the variable of interest (either the species concentrations or the temperature), D is the diffusivity (either mass or thermal), v is the average velocity, and R represents source or sink (generation or consumption) terms. Reactor design problems are generally interested only in the steady-state solution, in which the left hand side of (4.1) is zero. The local fluid velocity is calculated by the continuity equation:

$$\nabla \cdot (\rho v) = 0 \tag{4.2}$$

where ρ is the fluid density, which is a function of temperature, pressure and composition. Fluid density is calculated using an equation of state or a physical properties package, f^{PP}:

$$\rho = f^{PP}(T, P, \boldsymbol{z}) \tag{4.3}$$

where T is the temperature, P is the pressure, and $\boldsymbol{z}$ is the fluid composition. Finally, there is some pressure drop throughout the reactor bed which is determined using an empirical correlation, g_P (e.g., the Ergun equation [85]):

$$\nabla P = g_P(\rho, v, \mathbf{z}) \tag{4.4}$$

where $\mathbf{z}$ represents the set of design variables of the reactor (e.g., catalyst pellet diameter, bed width). The governing equations for the solid media in the reactor, which may include the reactor construction materials, catalyst particles, or membranes are given by:

$$\frac{\partial u^s}{\partial t} = \nabla \cdot (D^s \nabla u^s) + R^s \tag{4.5}$$

where u^s represents either the species concentrations or temperature in the solid phase, D^s is the mass or thermal diffusivity in the solid phase, and R^s represents sources or sinks in the

solid phase. The left-hand side of (4.5) is again zero when carrying out steady-state design calculations. Note that mass and heat flux between the solid and fluid phases, as well as reaction terms are embedded within the source and sink terms R and R^s.

These highly coupled, nonlinear partial differential equations usually cannot be solved analytically, and thus the spatial domain is discretized to approximate the (spatial) partial derivatives. The resulting steady-state model is a system of nonlinear algebraic equations. In an equation-oriented simulator, these equations would typically have to be solved using a sequential refinement method [208], solving approximate shortcut models first, and eventually the rigorous models. However, this chapter proposes a method for directly and simultaneously solving the rigorous reactor model equations along with the entire process flowsheet using pseudo-transient continuation (PTC).

Following the PTC strategy, only a simple system of equations is to be solved at the consistent initialization step. Then, as the pseudo-transient model is integrated in pseudo-time, the complete set of detailed model equations are gradually enforced. As described in Section 3.3.2, at the consistent initialization step, the source and sink terms in the reactor are set to zero using a continuation parameter $\hat{\alpha}$, and the concentration, velocity, and temperature profiles set to be uniform:

$$R^c = \hat{\alpha} R \tag{4.6}$$

$$R^{s,c} = \hat{\alpha} R^s \tag{4.7}$$

$$g^c = \hat{\alpha} g \tag{4.8}$$

where $\hat{\alpha}$ is described by (3.11)–(3.3.2) to be zero at $\hat{t} = 0$ and one at steady state. R^c, $R^{s,c}$, and g^c should then be substituted into the the model equations (4.1)–(4.5) for R, R^s, and

g, respectively. The mathematical equivalent of these conditions is given by:

$$u = u^0 \tag{4.9}$$

$$u^s = u^{s,0} \tag{4.10}$$

$$\nabla \cdot P = 0 \tag{4.11}$$

$$R = R^s = 0 \tag{4.12}$$

Additionally, the physical model equations (4.1)–(4.5) should be included in the model after replacing the time derivatives ($\frac{\partial u}{\partial t}$ and $\frac{\partial u^s}{\partial t}$) with pseudo-time derivatives ($\frac{\partial u}{\partial \hat{t}}$ and $\frac{\partial u^s}{\partial \hat{t}}$). The following initial conditions are provided:

$$u(\hat{t} = 0) = u^0 \tag{4.13}$$

$$u^s(\hat{t} = 0) = u^{s,0} \tag{4.14}$$

where u^0 and $u^{s,0}$ can simply be set to the fluid inlet conditions.

It can easily be seen that the consistent initialization step (at $\hat{t} = 0$) is very simple, as the concentration, temperature, and velocity profiles are all spatially uniform, and the source/sink terms are set to zero. Conversely, at steady state, the continuation parameter $\hat{\alpha}$ is one, and all the detailed phenomenological equations are accounted for in the model.

The challenge with this approach arises from the fact that the temperature must be high enough to "ignite" the reaction, and ensure that the reactor is stabilized. A pseudo-time "controller" can be implemented, which is formulated as an integral-only controller manipulating the inlet temperature to control the outlet temperature in pseudo-time.

$$\tau_I \frac{dT_{in}}{d\hat{t}} = T_{out}^{sp} - T_{out} \tag{4.15}$$

Here τ_I is the integral time constant, which should be at least as large as the dominant time constant of the reactor [229]. T_{out} is the reactor outlet temperature, and T_{out}^{sp} is the

"set point" temperature, which is either a parameter or an optimization decision variable during the design optimization. Additionally, the upstream units must compensate for the variability of the manipulated variable, the reactor inlet temperature, such that the energy balance is satisfied. Note that a PI control law can also be implemented; this may provide some convergence benefits, but presents the added challenge of defining a controller gain that ensures the stability of the resulting dynamical system.

4.2.2 Rigorous Pseudo-Transient Separation Models

In this section I give a general framework for rigorous pseudo-transient vapor-liquid separation models based on [193], which can be included in a process flowsheet along with the high-fidelity reactor models introduced in the previous section

4.2.2.1 Phase Equilibrium

The foundation for vapor-liquid separation systems modeling is the equilibrium flash calculation. In most process flowsheet modeling instances, pressure-enthalpy (PH) flash calculations are required (e.g., at every stage of a distillation column). Note that pressure-temperature (PT) flash calculations, where the pressure and temperature are fixed, are relatively easy to solve for vapor-liquid equilibrium (VLE), liquid-liquid equilibrium (LLE), or vapor-liquid-liquid equilibrium (VLLE) calculations [246, 290] either using Gibbs free energy minimization methods or equation-solving methods such as Rachford-Rice [151]. On the other hand, PH flash calculations, where the pressure and heat rate are fixed, are more difficult to solve and are subject to local-minimum solutions that are either unstable or physically unrealizable [246]. This is predominantly due to the fact that temperature is a source of significant nonlinearities in the phase equilibrium model (i.e., the fugacities, activity coefficients, and molar enthalpies are highly nonlinear functions of temperature).

To reliably solve pressure-enthalpy flash calculations, I propose using the pressure-temperature flash calculations provided by a physical properties package along with a pseudo-transient energy balance to calculate the temperature, as in (3.9). The pseudo-transient energy balance is formulated as:

$$\left(\frac{h^F}{T_0}\right)\tau_T\frac{dT}{d\hat{t}} = h^F + \frac{Q}{F} - \frac{L^I}{F}h^{L^I} - \frac{L^{II}}{F}h^{L^{II}} - \frac{V}{F}h^V \tag{4.16}$$

Equation (4.16) assumes that 3 phases are present (otherwise the molar flow rate of the phase not present is zero). τ_T is the phase equilibrium time constant (note that this should be a relatively small value compared to other time constants throughout the process, as in Figure 3.2). T is the flash temperature, F is the total inlet molar flow rate, h^F is the specific molar enthalpy of the feed, and Q is the heat rate. L^I, L^{II}, and V are the molar flow rates of liquid phase I, liquid phase II, and the vapor phase, respectively, and h^{L^I}, $h^{L^{II}}$, and h^V are the corresponding specific molar enthalpies. Stability considerations for this pseudo-transient energy balance are provided in [190]. Given an arbitrary physical property package, the molar enthalpies in each phase k, are nonlinear functions of pressure, temperature, and composition:

$$h^k = h^{PP,k}(T, P, \boldsymbol{z}^k) \tag{4.17}$$

where the composition in each phase, $\boldsymbol{z}^k = \boldsymbol{x}$ or $\boldsymbol{y}$, is an output of the phase equilibrium calculation. The complete pseudo-transient PH flash solution algorithm is then:

1. Given: total inlet enthalpy, heat rate, composition, and total flow rate
2. Set: initial temperature $T(\hat{t} = 0) = T^0$
3. Use the physical properties package available to perform a PT flash calculation
4. Calculate the molar enthalpies of each phase using equation (4.17).

5. Calculate the pseudo-time derivative for temperature from equation (4.16) explicitly.

6. Step forward in pseudo-time to update temperature

7. If the time derivative is zero, the pressure-enthalpy flash has been solved. Otherwise, return to step 2

This algorithm is depicted in Figure 4.2. In general, for a (VLLE) PH flash calculation, given inlet conditions F, $\boldsymbol{z}$, h^F, P, and Q, the calculated variables at the steady state solution are T, V, L^I, L^{II}, $\boldsymbol{y}$, $\boldsymbol{x}^I$, $\boldsymbol{x}^{II}$, h^V, h^{LI}, and h^{LII}.

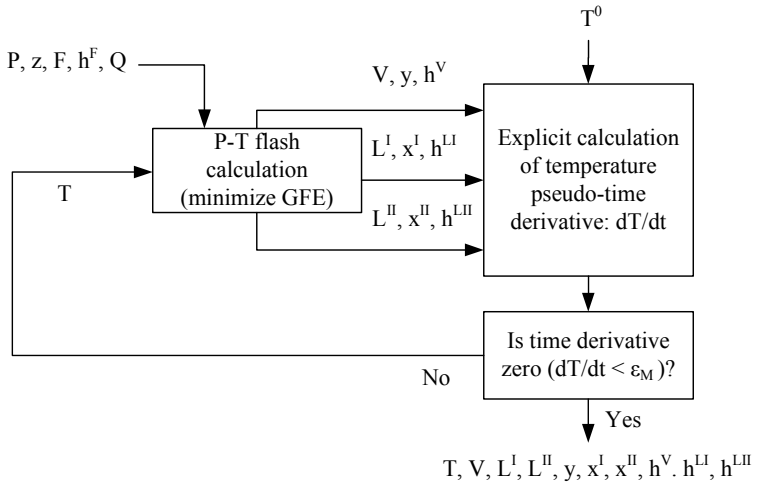

Figure 4.2: Diagram of the pseudo-transient pressure-enthalpy flash algorithm for up to 3 phases (VLLE). A Gibbs free energy (GFE) minimization subroutine calculates the phase equilibrium conditions at the current pressure and temperature. The temperature updates throughout the pseudo-transient integration with the dynamic energy balance (4.16).

Remark 4.1. It is important to note that this algorithm runs simultaneously with the entire flowsheet simulation. The user simply has to include equations (4.16), (4.17), and the PT flash call in the model equations and provide an initial condition for temperature, T^0 to obtain a solution of the PH flash calculation at the end of the pseudo-time integration.

4.2.2.2 Distillation Systems

As shown in Figure 4.3, distillation systems can be modeled as a system of unit operations including: (i) reboilers, (ii) condensers, (iii) feed stages, (iv) side draw stages, and (v) staged equilibrium sequences. These building blocks can be used to represent conventional distillation columns, stripping columns, absorption columns, multiple feed columns, side draw columns, or even dividing-wall columns [193]. All these units can employ embedded pseudo-transient PH flash calculations as described above to simplify their solution; however, the models are still highly coupled and difficult to solve (at the consistent initialization step) due to the connectivity equations setting the input material and energy flows to each equilibrium stage as the output variables of the adjacent stages.

To address this challenge, I implement a dynamic decoupling of the staged equilibrium sequences by reformulating the interconnectivity constraints as pseudo-transient differential equations. Specifically, for each equilibrium stage in each sequence, the pseudo-transient material exchange between stages is described by [190, 193]:

$$\tau_u \frac{dM_{i,j}}{d\hat{t}} = L_{i+1}x_{i+1,j} + V_{i-1}y_{i-1,j} - M_{i,j} \tag{4.18}$$

$$F_i = \sum_j M_{i,j} \tag{4.19}$$

$$z_{i,j} = \frac{M_{i,j}}{F_i} \tag{4.20}$$

Note that in this case I consider one liquid phase, but two liquid phases can be included with slight modification to the model equations.

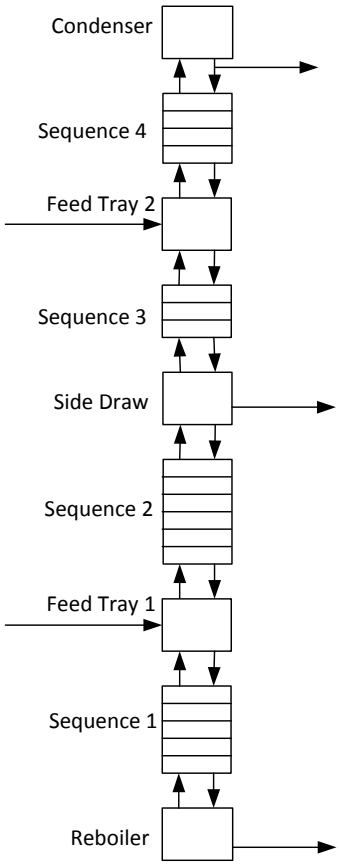

Figure 4.3: Representation of a distillation system as a flowsheet of generic building blocks [193]. This example has a reboiler, condenser, two feed stages, one side draw stage, and four staged equilibrium sequences.

Likewise, the pseudo-transient energy exchange is given by:

$$\tau_u \frac{dH_i}{d\hat{t}} = L_{i+1} h_{i+1}^L + V_{i-1} h_{i-1}^V - H_i \tag{4.21}$$

$$h_i^F = \frac{H_i}{F_i} \tag{4.22}$$

where $i \in I$ is the equilibrium stage (numbered from 1 at the bottom of the column to N at the top), $j \in J$ is the component, L is the liquid flow rate, and V is the vapor flow rate. The total molar flow rate of component j to stage i, $M_{i,j}$, and the total enthalpy of material streams to stage i, H_i, are given initial conditions, such that the phase equilibrium at each

stage can be solved independently at the consistent initialization step. The evolution in pseudo-time is defined such that at steady state, the pseudo-transient material and energy transfer equations (4.18)–(4.22) are equivalent to the mass and energy balances describing the steady-state operation. The dynamics of this pseudo-transient system are stable [190].

To allow for variable feed and side draw locations, as well as the total number of equilibrium stages included in the distillation column design, each staged sequence unit in the distillation flowsheet should consist of a variable number of equilibrium stages. Thus, the staged equilibrium sequence models incorporate stage bypass efficiencies [71, 193] to select the optimal number of stages and feed stage locations. A continuous variable, the bypass efficiency ϵ_i, is introduced for each stage i. The bypass efficiency is bounded between 0 and 1, with $\epsilon_i = 0$ corresponding to a full bypass of the stage and $\epsilon_i = 1$ corresponding to a fully functional equilibrium stage (no bypass). The liquid and vapor inlets from adjacent stages enter the equilibrium stage, and the bypass efficiency is applied at the stage outlets:

$$V_i = (1-\epsilon_i)V_{i-1} + \epsilon_i V_i^{eq} \tag{4.23}$$

$$V_i y_{i,j} = (1-\epsilon_i)V_{i-1}y_{i-1,j} + \epsilon_i V_i^{eq} y_{i,j}^{eq} \tag{4.24}$$

$$V_i h_i^V = (1-\epsilon_i)V_{i-1}h_{i-1}^V + \epsilon_i V_i^{eq} h_i^{V,eq} \tag{4.25}$$

At the liquid bypass and equilibrium mixing point, the balance equations are:

$$L_i = (1-\epsilon_i)L_{i+1} + \epsilon_i L_i^{eq} \tag{4.26}$$

$$L_i x_{i,j} = (1-\epsilon_i)L_{i+1}x_{i+1,j} + \epsilon_i L_i^{eq} x_{i,j}^{eq} \tag{4.27}$$

$$L_i h_i^L = (1-\epsilon_i)L_{i+1}h_{i+1}^L + \epsilon_i L_i^{eq} h_i^{L,eq} \tag{4.28}$$

The values of the bypass efficiencies are driven towards zero or one in the course of the optimization to avoid inefficient mixing between stages [70, 71]. An alternate proof that

the optimal designs include only binary values of the bypass efficiencies is provided in [193]. The total number of trays in a staged equilibrium sequence (NT) can then be calculated:

$$NT = \sum_{i=1}^{NMX} \epsilon_i \tag{4.29}$$

where NMX is the total number of stages (with or without bypass) in the equilibrium sequence model.

4.3 Case Study: Intensified Dimethyl Ether (DME) Process

I consider the DME synthesis process described in Appendix B of Analysis, Synthesis, and Design of Chemical Processes by Turton et al. [264]. The process flowsheet is shown in Figure 4.4, and the reactor system is described in the following sections. A pure methanol stream is combined with the recycle stream which is then vaporized in the reactor feed-effluent heat exchanger and sent to the reactor system where the methanol is dehydrated to form dimethyl ether. The outlet of the reactor is then cooled—if necessary—by a second heat exchanger and passed to the separation system, where the DME product and the methanol recycle stream are recovered. While distillation traditionally used to separate the DME product, it is very energy intensive, and thus a natural candidate for process intensification. Complex distillation configurations, such as Petlyuk columns, have gained significant interest lately for carrying out ternary separations, providing significant improvements in process energy efficiency and profitability [200]. To further reduce capital cost, the two columns of a Petlyuk system can be combined into a single column with a dividing-wall separating the stages above and below the feed and side draw, allowing for three high-purity products to be obtained in a single column [228].

Here, I consider an intensified design in which a dividing wall column (DWC) is used to carry out the three component separation [134, 135, 193]. The DME product is recovered

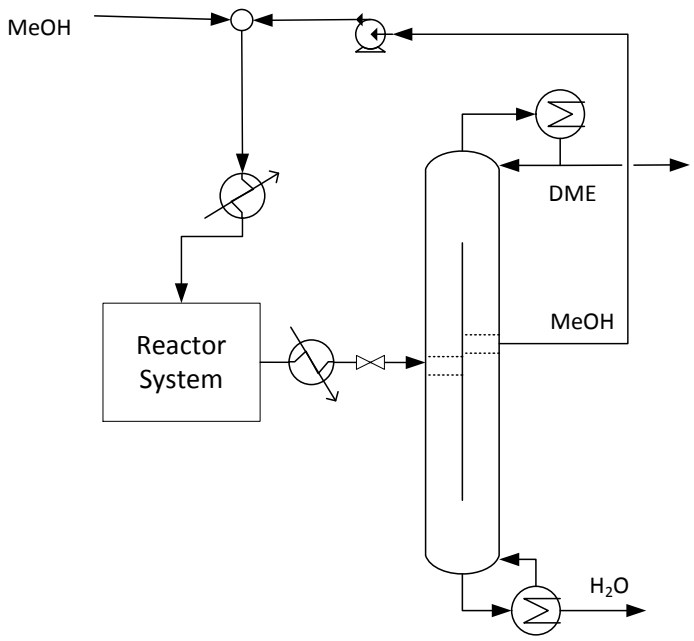

Figure 4.4: Diagram of the intensified DME process.

in the distillate, water in the bottoms, and methanol in the side stream. Pattison et al. [193] previously showed that significant savings are achieved when the intensified DWC design is implemented in place of the conventional two-column design; however, in all previous studies, simplified reactor models (e.g., lumped-parameter models, fixed conversion models) have been used to approximate the behavior of the unit. In this chapter I quantify the potential benefit of using a detailed first-principles reactor model at the process design stage by comparing the optimal solution to a process with a simple lumped-parameter reactor model, whereby the reactor is assumed to be well-mixed (similar to the results presented in the literature). The detailed first-principles reactor model accounts for axial variations in the reactor gas flow, radial variations in the catalyst pellets, and quenching.

I assume that only one reaction is occurring in the reactor below $400°C$, after which side reactions begin to occur:

$$2CH_3OH \rightarrow (CH_3)_2O + H_2O \tag{4.30}$$

The most widely accepted rate law for methanol dehydration on an alumina catalyst is given by Bercic and Levec [24, 25]:

$$R_r = \frac{kK_M^2(C_M^{s^2} - C_D^s C_W^s / K_{eq})}{(1 + 2(K_M C_M^s)^{(1/2)} + K_W C_W^s)^4} \tag{4.31}$$

where the subscripts $i = M, D, W$ refer to methanol, dimethyl ether, and water, respectively. The rate constant, k, and adsorption constants, K_i, follow the Arrhenius expression [25]:

$$k = k^0 \exp(\frac{-E_A}{RT}) \tag{4.32}$$

$$K_i = K_i^0 \exp(\frac{B_i}{T}) \tag{4.33}$$

The equilibrium constant is a function of temperature given by [97]:

$$\log(K_{eq}) = \frac{4019}{T} + 3.707\log(T) - 2.783 \times 10^{-3}T + 3.80 \times 10^{-7}T^2 - \frac{6.561 \times 10^4}{T^3} - 26.64 \tag{4.34}$$

The UNIQUAC package is used to calculate the physical properties and phase equilibria throughout the flowsheet model. The two reactor representations—a simple, lumped model as a basis for comparison, and a detailed distributed parameter model including intermediate quenching—are shown in Figures 4.5 and 4.6, respectively.

4.3.1 Lumped-Parameter Reactor Model

To quantify the benefit of including detailed, first-principles reactor models at the process design stage, I define a base case using a simple, lumped-parameter reactor model. A continuously stirred tank reactor (CSTR) model (rather than a fixed conversion reactor)

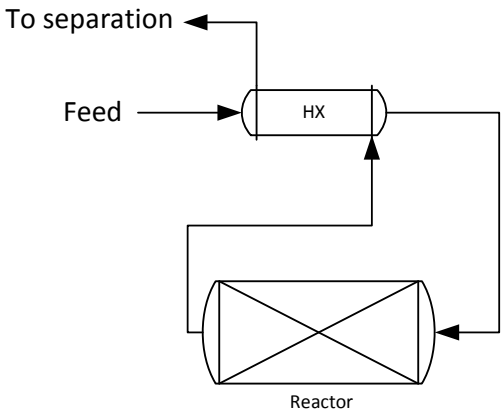

Figure 4.5: Diagram of the heat exchanger and reactor in the DME process with a lumped parameter reactor model.

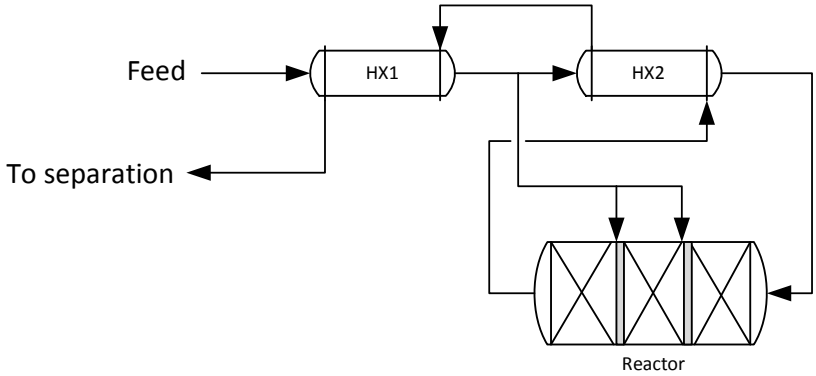

Figure 4.6: Diagram of the heat exchanger and reactor in the DME process with the detailed distributed parameter reactor model.

was selected, given that, (i) the reaction kinetics are known and (ii) such models are readily available in any basic flowsheeting software, and are the likely choice for modeling the system under consideration. A pseudo-transient CSTR model was adapted from [190].

The steady-state mass balance equation for a component i in a CSTR is given by:

$$0 \quad = \quad F(C_{i,in} - C_i) - r_i \tag{4.35}$$

where F is the total volumetric flow rate through the reactor, $C_{i,in}$ is the inlet concentration of component i, and C_i is the concentration of component i in the reactor. The reaction

term r_i is the overall rate of consumption and/or generation of component i in the reactor.

The energy balance is again written as a differential equation in pseudo-time, and the heat of reaction is treated as a sink/source term using the continuation parameter $\hat{\alpha}$:

$$\tau_T \frac{dT_r}{d\hat{t}} = (T_{r,in} - T_r) - \frac{\hat{\alpha} R_r \Delta H}{\rho c_p F} \tag{4.36}$$

where T_r is the temperature in the reactor, $T_{r,in}$ is the temperature of the inlet flow, ρ and c_p are respectively the average density and heat capacity inside the reactor, and ΔH is the heat of reaction. The reaction rate, R_r is calculated from (4.31).

Since the reactor is approximated as a well-mixed volume, the entire reactor must be at a temperature low enough to avoid the advent of side reactions. The well-mixed lumped parameter reactor model cannot (and need not) consider the intermittent quenching obtained by splitting the feed stream, as in Figure 4.6. Rather, the flowsheet is arranged with the entire feed stream entering the simple reactor model as a single inlet, and only one heat exchanger is required to vaporize the reactor inlet. In this case, the "Reactor System" of the DME production flowsheet (bottom-left of Figure 4.4) is replaced by the units shown in Figure 4.5.

4.3.2 Spatially Distributed Reactor Model

I consider a gas-phase plug-flow reactor with inter-stage quenching to prevent undesired side reactions. Each reaction stage in the reactor is modeled as an axially dispersed plug-flow reactor, and complete mixing is assumed at the quenching points. It is further assumed that radial variation in the bed is neglible and that diffusion into the catalyst pellets can be described by Fick's law.

The pseudo-transient gas phase material balance is derived by applying an arbitrary time constant τ_d to (4.1) written in terms of component concentrations (since the reaction is equimolar):

$$\tau_d \epsilon \frac{\partial C_i}{\partial \hat{t}} = -\frac{\partial (v C_i)}{\partial z} + \frac{\partial}{\partial z}\left(D_{mix,i} \frac{\partial C_i}{\partial z} \right) + \epsilon a_e k_{g_i} (C_i^s(d_p/2) - C_i) \tag{4.37}$$

where ϵ is the void fraction, C_i is the concentration of component i in the gas phase, $z \in [0, L]$ is the axial dimension in the reactor, v is the gas velocity, and $D_{mix,i}$ is the diffusivity of component i in the fluid mixture. The driving force for mass transfer between the solid and gas phase is the concentration difference between the catalyst surface $C_i^s(d_p/2)$ and the gas phase. The mass transfer coefficient for component i is k_{g_i}, and a_e is the effective area (per unit volume) given by $a_e = \frac{6(1-\epsilon)}{\epsilon d_p}$, where d_p is the catalyst pellet diameter.

The pseudo-transient energy balance in the gas phase can similarly be derived by applying equation (4.1) to temperature and is given by:

$$\tau_1 \epsilon c_p^g \frac{\partial T}{\partial \hat{t}} = -\frac{\partial (v c_p^g T)}{\partial z} + V_m \frac{\partial}{\partial z}\left(\lambda \frac{\partial T}{\partial z} \right) + V_m a_e h_f (T^s - T) \tag{4.38}$$

where c_p^g is the molar heat capacity of the gas mixture, T and T_s are the gas and solid phase temperatures, respectively, V_m is the molar volume of the gas mixture, λ is the thermal conductivity of the gas mixture, and h_f is the heat transfer coefficient. Note that the continuation parameter, $\hat{\alpha}$, is not included in the gas phase equations (4.37)-(4.38) because the initial conditions are set such that the source/sink terms are zero at the consistent initialization ($C_i^s(d_p/2, \hat{t} = 0) - C_i(\hat{t} = 0) = 0$ and $T^s(\hat{t} = 0) - T(\hat{t} = 0) = 0$).

The pressure drop is modeled using the Ergun equation [85]:

$$P(z) = P^0 - \hat{\alpha} \int_0^z \left(\frac{150(1-\epsilon)^2 \mu v}{d_p^2 \epsilon^3} + \frac{1.75(1-\epsilon)\rho^g v^2}{d_p \epsilon^3} \right) dz' \tag{4.39}$$

where $P(z)$ is the pressure along the reactor, P^0 is the inlet pressure, μ is the viscosity of the gas mixture, and ρ^g is the gas phase density. I assume a static continuity equation:

$$\frac{\partial(\rho^g v)}{\partial z} = 0 \tag{4.40}$$

The binary diffusivity between two components, i and j is given by [95]:

$$D_{i,j} = 0.01T^{(7/4)}\frac{\sqrt{(1/M_i)+(1/M_j)}}{P((\sum V_{d,i})^{(1/3)}+(\sum V_{d,j})^{(1/3)})^2} \tag{4.41}$$

where $V_{d,i}$ and $V_{d,j}$ are the diffusion volumes of components i and j, respectively (tabulated in [95]), and M_i and M_j are the molar masses of components i and j. The diffusivity of each component i in the mixture is given by:

$$D_{mix,i} = \frac{(1-y_i)}{\sum_j \frac{y_j}{D_{i,j}}} \tag{4.42}$$

where y_i is the mole fraction of component i. The solid-fluid heat transfer coefficient, h_f, is determined by [272]:

$$Nu = 2 + 1.1Re^{(3/5)}Pr^{(1/3)} \tag{4.43}$$

where the Nusselt number is defined as $Nu = \frac{h_f d_p}{\lambda}$, the Reynolds number is $Re = \frac{\rho^g u d_p}{\mu}$, and the Prandtl number is $Pr = \frac{c_p^g \mu}{\lambda}$. The solid-fluid mass transfer coefficient is given by [87]:

$$Sh_i = 2 + 1.1Re^{(3/5)}Sc_i^{(1/3)} \tag{4.44}$$

where the Sherwood number is $Sh_i = \frac{k_{g_i} d_p}{D_{mix,i}}$, and the Schmidt number is $Sc_i = \frac{\rho^g \mu}{D_{mix,i}}$.

The pseudo-transient material balance for component i in the catalyst pellets is derived by applying the same time constant τ_d to (4.5) in the pellets:

$$\tau_d \frac{\partial C_i^s}{\partial \hat{t}} = D_{eff,i}\left(\frac{\partial^2 C_i^s}{\partial r^2} + \frac{2}{r}\frac{\partial C_i^s}{\partial r}\right) + \hat{\alpha}\rho^s \nu_i R_r \tag{4.45}$$

where C_i^s is the concentration of component i in the catalyst, $r \in [0, \frac{d_p}{2}]$ is the radial dimension, and $D_{eff,i}$ is the diffusivity of component i within the catalyst (given by $D_{eff,i} = D_{mix,i}\epsilon_p/\gamma$, where ϵ_p and γ are the catalyst porosity and tortuosity, respectively). ρ^s is the catalyst density, ν_i is the stoichiometric coefficient of component i, and R_r is the rate of reaction. Due to the relatively small value of the Biot number ($Bi = \frac{h_f d_p}{6\lambda_{eff}} \approx 0.1$), it is assumed that there are no radial temperature variations in the catalyst pellets. The pseudo-transient heat balance in the catalyst pellets is thus given by:

$$\tau_2 \rho^s c_p^s (1-\epsilon) \frac{dT^s}{d\hat{t}} = \epsilon a_e h_f (T - T^s) - \hat{\alpha}(1-\epsilon)\eta\rho^s(-\Delta H_r)R_r \tag{4.46}$$

where c_p^s is the heat capacity of the catalyst, ΔH_r is the heat of reaction, η is the effectiveness factor, which is given by:

$$\eta = \frac{\int R_r dV}{V\ R_r|_{\frac{d_p}{2}}} \tag{4.47}$$

Assuming a spherical catalyst pellet, the effectiveness factor is:

$$\eta = \frac{3\int_0^{\frac{d_p}{2}} R_r r^2 dr}{d_p^3\ R_r|_{\frac{d_p}{2}}} \tag{4.48}$$

To maintain a temperature low enough to avoid side reactions in the detailed reactor, the feed stream is split, and the reactor is quenched at two points. The feed stream is fed through two heat exchangers before it reaches the reactor inlet, and a portion of the feed is taken after the first heat exchanger and used to quench the reactor at two intermittent points (see Figure 4.6). Via this quenching strategy, the reactor is cooled at two points, allowing operation at higher temperatures throughout the reactor while still avoiding side reactions.

To stabilize the exothermic reaction throughout the pseudo-time horizon, a "control" strategy similar to (4.15) is implemented to control the outlet temperature from each reaction

segment ($T_{out,1}$, $T_{out,2}$, and $T_{out,3}$). In the first segment, the manipulated variable is the heat provided by the first heat exchanger (Q_{HX1}):

$$\tau_I \frac{dQ_{HX1}}{d\hat{t}} = T^{sp}_{out,1} - T_{out,1} \tag{4.49}$$

In the second and third segments, the manipulated variables are the fraction (ξ_1 and ξ_2) of the methanol feed sent to each quenching point.

$$\tau_I \frac{d\xi_1}{d\hat{t}} = T^{sp}_{out,2} - T_{out,2} \tag{4.50}$$

$$\tau_I \frac{d\xi_2}{d\hat{t}} = T^{sp}_{out,3} - T_{out,3} \tag{4.51}$$

where the setpoints for each outlet temperature ($T^{sp}_{out,1}$, $T^{sp}_{out,2}$, and $T^{sp}_{out,3}$) are fixed to the maximum allowable temperature, $390°C$.

The intermittent quenching points are approximated as well-mixed, with no radial variation. In effect, the entire reactor is equivalent to three smaller reactors, split at the quenching points, for which the inlet of each reactor is the outlet of the previous reactor combined with a portion of the methanol feed added to quench the temperature. In the calculations carried out with this detailed model, the "Reactor System" in Figure 4.4 is replaced by the structure in Figure 4.6 using the model as described above.

Consistent Initialization of the Reactor Models. The pseudo-transient mass and energy balances in the gas phase are initialized with the initial conditions:

$$C_i(\hat{t} = 0) = C_{i,in} \tag{4.52}$$

$$T(\hat{t} = 0) = T_{in} \tag{4.53}$$

Similarly, the pseudo-transient catalyst pellet material and energy balances are initialized with the inital conditions:

$$C_i^s(\hat{t}=0) = C_{i,in} \tag{4.54}$$

$$T^s(\hat{t}=0) = T_{in} \tag{4.55}$$

Boundary conditions are applied to ensure symmetry in the catalyst pellet and equal component mass fluxes at the catalyst-gas boundary.

4.3.3 Dividing-Wall Column Model

The separation system for the intensified DME synthesis process flowsheet consists of a dividing-wall distillation column (DWC). Using the general rigorous separation system modeling framework established in Section 4.2.2, a DWC model is easily constructed using the unit operation building blocks described. The model consists of a reboiler, total condenser, feed stage, side draw stage, and six staged equilibrium sequences—each with a variable number of equilibrium stages modeled using stage bypass efficiencies. An additional constraint is included to ensure the feasible design of the DWC, i.e., the number of stages on either side of the dividing-wall is the same. The DWC model has six design degrees of freedom in addition to the number of equilibrium stages in each sequence: (i) the reflux ratio, (ii) the liquid split at the top of the dividing-wall, (iii) the side draw flow rate, (iv) the vapor split at the bottom of the dividing-wall, (v) the reboiler heat rate, and (vi) the pressure in the reboiler or condenser. I assume a linear, fixed pressure drop is provided.

4.3.4 Dimethyl Ether (DME) Flowsheet Optimization

For process optimization, the objective function is defined as the sum of capital and operating expenses. The tradeoff between the number of equilibrium stages in the distillation

column and the reboiler heat duty is based on example MT5 in Viswanathan and Grossman [271]. The cost of cooling water and the reactor volume are also penalized in the flowsheet design objective function, which is:

$$OBJ = 63.88 \times 10^{-6}(Q_{reb} + Q_{mps}) + NT + 20 \times 10^{-6}(Q_{cond} + Q_{chill}) + 0.5(V_r) \quad (4.56)$$

where Q_{reb} and NT are respectively the reboiler duty and number of equilibrium stages in the dividing-wall column, and Q_{mps} is the medium pressure steam heat duty (MW). Q_{cond} is the condenser heat load and Q_{chill} is the chiller cooling water heat load. All heat loads are in MW. V_r is the reactor volume in cubic meters.

The following product quality and process operating constraints are enforced:

- The DME product flow rate must be greater than 128 mol/s
- The product stream must have a purity of at least 99.9 mol% DME
- The recycle stream should be no more than 5.0 mol% water
- The reboiler temperatures cannot exceed 465 K
- The condenser temperatures cannot be below 315 K
- The reactor temperature cannot exceed 663 K (390°C)

The optimization decision variables are shown in Table 4.1. As expected, the detailed reactor model has more design decision variables than the simple reactor model. The detailed reactor model design decisions include the reactor dimensions, catalyst pellet diameter (to reflect the tradeoff between the catalyst effectiveness and pressure drop), and the locations of the quenching points. Note that the reactor temperature is not a design decision in the

Table 4.1: DME process optimization decision variables (NMX is the number of potential stages in the DWC).

Decision Variable	Simple Reactor	Detailed Reactor	DWC
1	Reactor Length	Reactor Length	Reflux Ratio
2	Reactor Diameter	Reactor Diameter	Side Draw Fraction
3	Reactor Temperature	Quench Point 1	Vapor Split
4	Chiller Temperature	Quench Point 2	Liquid Split
5		Chiller Temperature	Reboiler Pressure
6		Quench Stream Temperature	Reboiler Duty
7		Catalyst Pellet Diameter	Bypass Efficiencies
Total	4	7	6 + NMX

detailed case because the temperatures are controlled, such that the exit temperature and the temperatures before quench points are all at the maximum allowed value.

The complete flowsheet with the simplified reactor model features 10 design decision variables, while the complete flowsheet with the detailed reactor model features 13 design decision variables. Additionally, I included 95 potential equilibrium stages for the design of the DWC in each model, thereby adding 95 more design decision variables (stage bypass efficiencies) to the optimization calculation. Both models were solved with 8 constraints, resulting in a system of 5152 equations for the simple reactor flowsheet and 14494 equations for the detailed reactor flowsheet. The optimization problems were solved using the reduced sequential quadratic programming algorithm (NLPSQP) in gPROMS [204]. Each flowsheet was optimized on an Intel Core i7 processor running at 3.40 GHz, and the optimal solutions were found in 1.22 hours for the simple reactor case and 5.0 hours for the detailed reactor case.

4.3.5 Results and Discussion

The full flowsheet was optimized with both the lumped-parameter reactor model and the detailed distributed parameter model, and the dividing wall column design results are presented in Table 4.2 and Figure 4.7. The reactor design results are presented in Table 4.3.

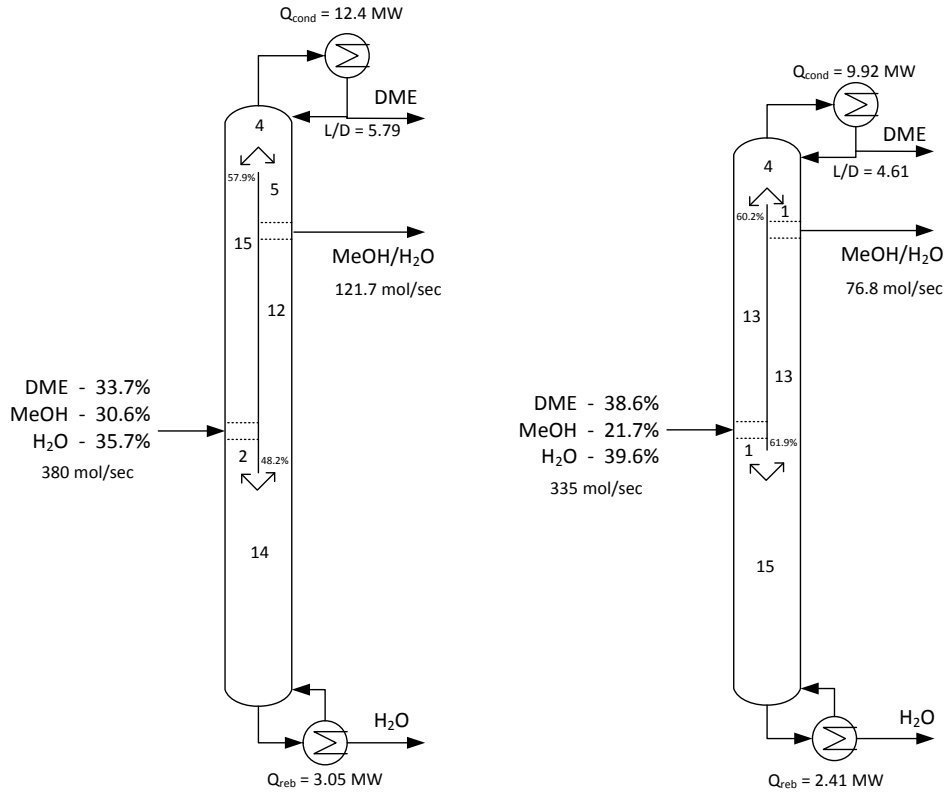

Figure 4.7: Comparison of the DWC structures determined from the optimization calculations with the simple (left) and detailed (right) reactor models.

The most notable differences between the two DWC designs stem from the detailed reactor model optimization resulting in a higher conversion of methanol to DME: 69% of feed methanol was converted to product in the simple reactor model, and 78% of feed methanol was converted to product in the detailed reactor model. As a result, both the side draw flow rate from the DWC and the utilities required to distill DME to the required purity are decreased. The differences in the capital cost and operating cost contributions between the two models are shown in Table 4.4.

Both the detailed reactor model and the simple reactor model operate at their upper temperature bound (390°C). For the simple, lumped-parameter reactor model, the whole reactor operates at this temperature. For the detailed, distributed parameter reactor model,

Table 4.2: DME process dividing-wall column results.

	Simple Reactor Flowsheet	Detailed Reactor Flowsheet
Feed flow (mol/s)	380.0	335.0
Feed composition (mol %)	30.6% MeOH	21.7% MeOH
	33.7% DME	38.6% DME
	35.7% H_2O	39.6% H_2O
Reboiler pressure (bar)	12.5	12.5
Condenser pressure (bar)	12.2	12.2
Reflux ratio	5.793	4.606
Distillate flow rate (mol/s)	128.0	128.0
Distillate composition (mol %)	0.1% MeOH	0.1% MeOH
	99.9% DME	99.9% DME
	0% H_2O	0% H_2O
Side draw flow rate (mol/s)	121.7	76.8
Side draw composition (mol %)	95.6% MeOH	94.7% MeOH
	0.2% DME	0.2% DME
	4.2% H_2O	3.2% H_2O
Bottoms flow rate (mol/s)	130.3	130.2
Bottoms composition (mol %)	0% MeOH	0% MeOH
	0% DME	0% DME
	100% H_2O	100% H_2O

Table 4.3: DME process reactor results.

	Simple Reactor	Detailed Reactor
Reactor Length (m)	4.1	6.8
Reactor Diameter (m)	4.1	3.0
Reactor Temperature (K)	663	-
Quenching Point 1 (m)	-	1.1
Quenching Point 2 (m)	-	1.7
Chiller Temperature (K)	422.7	421.8
Catalyst Pellet Diameter (mm)	-	1.5

the reactor reaches the temperature upper bound at each quenching point (the cooling is instantaneous because the quenching points are assumed to be well-mixed). The temperature and concentration profiles throughout the detailed reactor are shown in Figure 4.8.

An efficiency rating, defined as the process heat energy (Q_{mps}+Q_{reb}) used per kilogram of DME product, is typically used to quantify the efficiency of the process. The optimized design with the lumped-parameter model represents the best design found in previous literature [193], for which the efficiency rating is 0.501 kWh/kg. The optimized design with

Table 4.4: DME process optimization cost results.

	Simple Reactor	Detailed Reactor
Q_{reb} (MW)	3.05	2.41
Q_{mps} (MW)	7.55	5.88
NT	37	35
Q_{cond} (MW)	12.4	9.92
Q_{chill} (MW)	0	0
V_r (m^3)	449	464
J (AU)	1427.8	1238.6

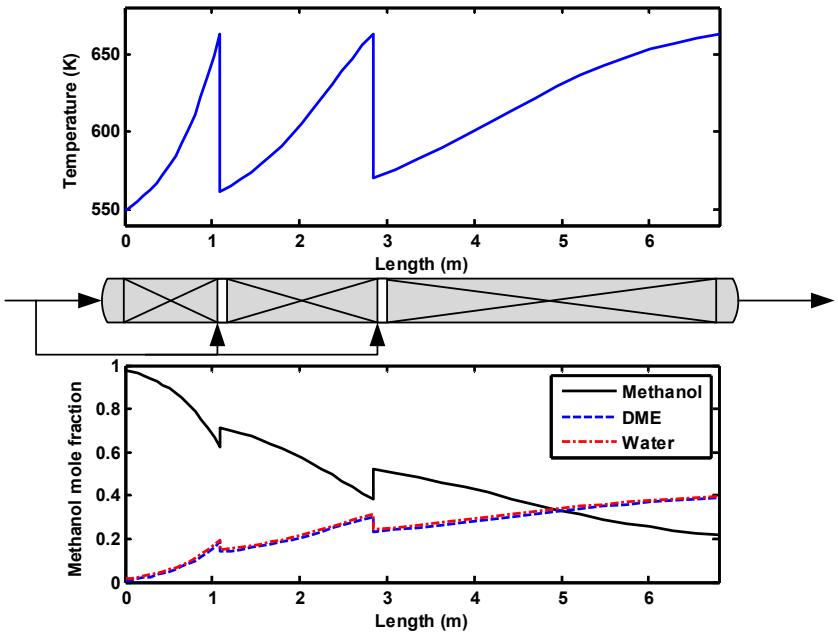

Figure 4.8: Temperature and mole fraction profiles throughout the optimized detailed reactor model. The quenching points are approximated as well-mixed plugs with no radial variation.

the detailed distributed model achieves a better efficiency rating because of the previously described reduction in utilities usage, and the efficiency rating of the optimized design is 0.392 kWh/kg, representing a 21.7% energy savings per unit of product. Additionally, the intensified process design with the lumped-parameter reactor model represents an 11.7% savings compared to the optimal reactive DWC configuration reported by Kiss and Suszwalak [134], and the corresponding design with the detailed reactor represents a 29.5% savings.

4.4 Summary

This chapter presented a transparent and unified framework for simulation and optimization of process flowsheets containing detailed, multiscale models of complex, intensified unit operations. A novel pseudo-transient representation was described for reactive systems governed by partial differential equations. This representation is suitable for embedding in previously-developed pseudo-transient process modeling, simulation and optimization framework, resulting in a multi-resolution representation of process flowsheets. While the developments were presented within the canonical reaction/separation/recycle integrated process paradigm, they are in effect general and can be extended and applied to any process structure including alternative, intensified designs.

Chapter 5

Simulation and Optimization of Process Flowsheets Incorporating Detailed Spiral-Wound Multistream Heat Exchanger Models[†]

In this chapter, I examine the incorporation of heat exchanger geometry into process flowsheet modeling, simulation, and optimization. I focus on spiral-wound heat exchangers (SWHEs)—a practically important class of equipment that is often heuristically designed, as described in Chapter 2. I include the geometric design of the SWHE into the flowsheet specifications and calculate the stream phases, pressures, and heat transfer coefficients along the exchanger based on the SWHE geometry using industry-accepted correlations. The models are formulated using the pseudo-transient approach such that (for the first time, to my knowledge) SWHE geometry can be simultaneously optimized with the process flowsheet parameters. I demonstrate this framework via the optimization of the PRICO®[‡] natural gas liquefaction process. The results confirm the challenges identified in Section 2.3.2, showing that previous attempts at optimizing the PRICO process may in fact result in sub-optimal, or even infeasible solutions, when the SWHE geometry is considered. The presentation in this chapter follows closely the developments in Tsay et al. (2017) [260].

[†]The contents of this chapter are largely based on the following publication: C. Tsay, R.C. Pattison, and M. Baldea. Equation-oriented simulation and optimization of process flowsheets incorporating detailed spiral-wound multistream heat exchanger models. AIChE J., 63(9):3778–3789, 2017. C.T. is the primary author of the manuscript.

[‡]PRICO® is a registered trademark of Black & Veatch holding company.

5.1 Multistream Heat Exchanger Modeling and Optimization

Multistream heat exchangers are an *intensified* unit used for cooling, condensing, vaporizing, and/or liquefying, commonly implemented in cryogenic processes (e.g. air separation, liquefaction, refrigeration). These energy-heavy processes rely on tight heat integration for economic efficiency and can greatly benefit from careful process design and optimization. In particular, recent interest in natural gas as a major fuel source has increased the need for economical means for natural gas liquefaction for long-range transportation. Liquefaction processing makes up around 52% of the final cost of liquefied natural gas (LNG) [116], leading to extensive efforts for improving energy efficiency. The multi-stream spiral-wound heat exchanger, allowing for thermal contact between multiple streams in a single unit, is most commonly selected for this application, with a refrigerant evaporating on the shell side in downward flow and hot streams circulating upwards on the tube side [39, 180, 241].

The incorporation of multistream heat exchanger (MHEX) units into process flowsheet modeling requires models that can accurately predict phase transitions and the corresponding change of physical properties. However, simulating and optimizing the resulting flowsheet models is challenging because the stream phases are often unknown *a priori*, and the phase boundaries (bubble point and dew point) are dependent on the stream composition and pressure, which change between flowsheet optimization iterations. The MHEX models available in commonly-used process simulation software are typically limited to solving a set of energy balance equations with the purpose of determining the outlet conditions of one stream, with all other inlet and outlet parameters known [122]. These conditions may change as the flowsheet variables change, and, in turn, the structure of the energy balance equations may change as well. While this is acceptable in sequential-modular flowsheeting software, it poses considerable difficulties for equation-oriented process simulation and optimization.

Previous approaches to equation-oriented modeling and optimization of MHEXs involving phase changes have explored the use of a superstructure [108], in which a network of two-stream heat exchangers equivalent to the MHEX is developed. Although such models allow for phase change calculations (provided the inlet and outlet stream states are known beforehand), the resulting non-convex, mixed-integer formulations are difficult to use in process optimization. Later, Kamath et al. [122] proposed an extension to the seminal heat integration approach of Duran and Grossmann [76] that uses a disjunctive programming-based formulation to capture phase change effects and variable heat capacities throughout the MHEX. The proposed model is pinch-based, ensuring a minimum temperature driving force during flowsheet optimization, and uses a smoothing function [13] to facilitate numerical solution. More recently, Cao et al. [46] proposed dealing with cases where only one stream undergoes phase change by dividing the exchanger into three segments, with boundaries defined such that the stream in question is in a single phase regime in each segment.

In addition to handling phase changes, MHEX models should ideally consider the required heat transfer area, which in turn provides information on device cost and construction. The bulk of literature regarding modeling and optimization of multi-stream heat exchangers considers the calculation of the size of the heat exchanger only after successful computation of the output stream parameters, effectively ignoring the dependence of stream states on the exchanger design. The performance of the MHEX, defined in terms of the pressure losses and heat transfer parameters, is generally assumed to be fixed during flowsheet optimization, and a suitable heat exchanger must be designed to fit these process specifications (see Section 2.3.2). Thus, the heat exchanger is typically designed iteratively, relying on the expertise of an experienced engineer. The engineer must attempt to meet flowsheet design specifications by satisfying a compromise between pressure drops and thermal exchange performance through many iterative trials until a reasonable design is obtained [112, 214].

5.2 Detailed Spiral-Wound Heat Exchanger Model

In this section, I present a detailed model for a (multistream) spiral-wound heat exchanger, or SWHE, based on industry-accepted, semi-empirical pressure drop and heat transfer correlations reported in the literature. The developments considerably extend the work of Pattison and Baldea [192], which provided a pseudo-transient formulation of the energy balance equations for a general MHEX without geometric or spatial considerations.

5.2.1 Geometry and Governing Equations

The geometry of a spiral-wound heat exchanger (SWHE) is characterized by an extensive set of parameters, which can be considered as decision variables for equipment/process design optimization. Figure 5.1 shows the tube bundles of a SWHE under construction.

Figure 5.1: A spiral-wound heat exchanger under construction.
Photograph from The Linde Group, freely available at
`http://www.lindeus.com/en/news_and_media/image_library/index.html`. Accessed 11/1/2016.

Figure 5.2 displays the interior structure and parameter descriptions for a SWHE. The space between the outer and inner shells of the exchanger is occupied by coils fixed by spacing

bars between each radial layer. Fluid on the shell side of the SWHE flows countercurrent to the fluid(s) on the tube side. In Figure 5.2, I assume that there are 2 tube-side streams, where stream 1 has a smaller coil diameter $d_{coil,1}$ and stream 2 has a larger coil diameter $d_{coil,2}$ (note that $d_{coil,2}$ denotes the outermost coil diameter if there are multiple layers of tubes for a particular stream). Additionally, the streams on the tube side (i.e., streams 1 and 2) are split into many tubes which can be staggered both laterally, where each tube has the same coil diameter (e.g., stream 2), and/or radially, where each tube has a different coil diameter (e.g., stream 1). Typically each tube-side stream involves both arrangements, resulting in tens to hundreds of tubes for each stream (see Figure 5.1 for an illustration of a stream being distributed into many tubes).

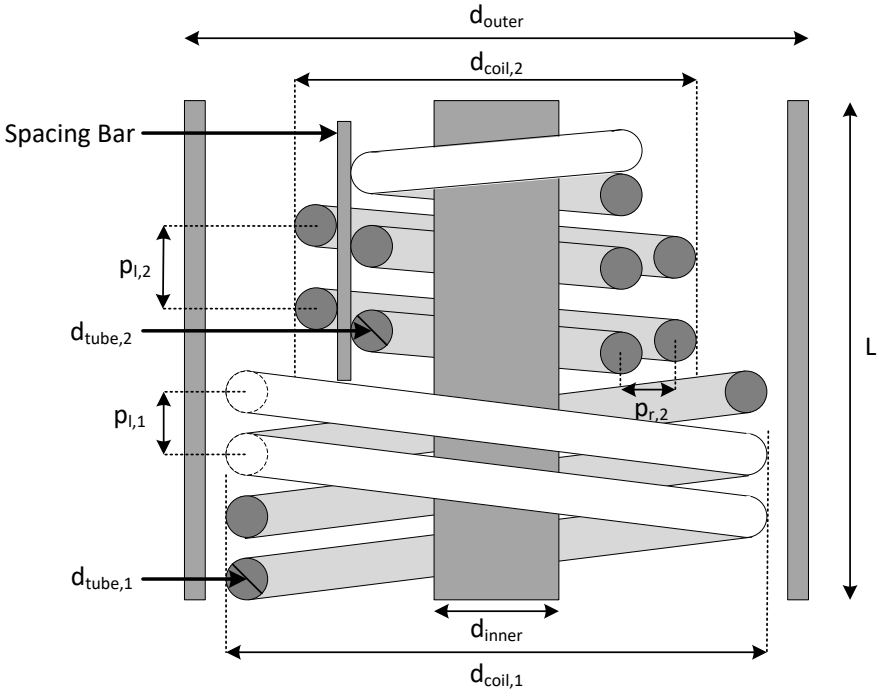

Figure 5.2: Axial cross section of a spiral-wound multistream heat exchanger.

The key geometric design variables include the exterior dimensions of the SWHE: the vertical length of the heat exchanger L, the outer diameter d_{outer}, and the inner diameter

d_{inner}. Each tubular coil in the exchanger is described in terms of an outer diameter d_{coil}, and the outer diameter of the coiled tube itself d_{tube}. Each coil is further characterized by a longitudinal pitch p_l representing the longitudinal distance between the center of parallel tubes and a radial pitch p_r representing the radial distance between the center of parallel tubes (see Figure 5.2). Finally, each stream has a corresponding tube thickness, l_i. Each of these geometric design decisions cannot be considered independently, as all contribute significantly to the overall performance of the heat exchanger.

I begin by assuming that variations in the stream properties only occur in the axial dimension in the heat exchanger. The specific stream enthalpies h in the exchanger are thus only functions of the axial dimension $z \in [0, L]$, and the steady-state energy balance for hot and cold streams is given by:

$$-F_i \frac{\partial h_i}{\partial z} = \sum_{j \in \mathcal{C}} U \bar{A}_{i,j}(z)(T_i(z) - T_j(z)) \tag{5.1}$$
$$F_j \frac{\partial h_j}{\partial z} = \sum_{i \in \mathcal{H}} U \bar{A}_{i,j}(z)(T_i(z) - T_j(z))$$

where $\mathcal{H}$ is the set of hot streams, $\mathcal{C}$ is the set of cold streams, and $i \in \mathcal{H}$ and $j \in \mathcal{C}$ respectively denote the i^{th} hot stream and the j^{th} cold stream. F_s is the molar flow rate of stream s (where $s \in S$ and $S = \mathcal{C} \bigcup \mathcal{H}$), h_s is the specific molar enthalpy of stream s, and T_s is the temperature of stream s. $U\bar{A}_{i,j}(z)$ is the overall heat transfer coefficient multiplied by differential area ($\bar{A} = \frac{dA}{dz}$) between the i^{th} hot stream and the j^{th} cold stream.

5.2.2 Phase Equilibrium Calculations

Each stream in the SWHE can change phase as it passes through the heat exchanger. The heat transfer and pressure drop correlations are dependent upon the phase of each stream, and, in the two-phase regime, the correlations also consider the composition of each phase. Under the assumption that the stream phases are in equilibrium, flash calculations

are required to compute the vapor fraction (X_s) and the liquid and vapor mole fractions ($\boldsymbol{x}_s$ and $\boldsymbol{y}_s$, respectively) of each stream $s \in S$. The developments in this chapter only require PT flash calculations (i.e., based on T_s, P_s, and the overall mole fractions $\boldsymbol{z}_s$), which, as described in Section 4.2.2.1, are easily performed using either a Gibbs energy minimization or equation-solving approach via a physical properties package or equation of state.

5.2.3 Pseudo-Transient Temperature Calculations

The governing heat balance equations (5.1) are used to compute the enthalpy of each stream. The temperature of each stream must be calculated from the molar enthalpy of the stream along the length of the heat exchanger.

$$0 = h_s(z) - h^{PP}(T_s(z), P_s(z), \boldsymbol{z}_s) \tag{5.2}$$

However, the relationship between temperature and specific enthalpy is a piecewise, nonlinear, and implicit function:

$$h^{PP}(T_s(z), P_s(z), \boldsymbol{z}_s) = \begin{cases} h^L(T_s(z), P_s(z), \boldsymbol{z}_s) & T_s \leq T_{bub} \\ h^{2p}(T_s(z), P_s(z), \boldsymbol{z}_s) & T_{bub} < T_s < T_{dew} \\ h^V(T_s(z), P_s(z), \boldsymbol{z}_s) & T_s \geq T_{dew} \end{cases} \tag{5.3}$$

where $T_s(z)$,$P_s(z)$, and $\boldsymbol{z}_s$ are the temperature, pressure, and composition of process stream s along the axial domain (z) of the heat exchanger. Superscripts L, $2p$, and V respectively denote the liquid, two-phase, and vapor phase regimes. I again assume that a physical properties package is available to compute stream molar enthalpy, as in (4.17). Furthermore I assume that the physical properties package can compute relevant derivatives of h^{PP} as a function of the stream properties. Note that such features are available through several commercial software packages.

Solving (5.2) is challenging when the phase of the stream is unknown *a priori*. However, a pseudo-transient reformulation of can be used to reliably solve for the steady state

stream temperatures $T_s(z)$ [192]. The reformulation requires that the stream temperatures become dynamic state variables in $\hat{t}$. Unlike in (3.9), the pseudo-transient temperature dynamics are proportional to the residual of (5.2) instead of the energy balance equation:

$$(\frac{h_{ref}}{T_s^0})\tau_e \frac{\partial T_s(z)}{\partial \hat{t}} = h_s(z) - h^{PP}(T_s(z), P_s(z), \boldsymbol{z}_s) \tag{5.4}$$

where $\frac{h_{ref}}{T_s^0}$ is a scaling factor to ensure that the units are consistent, and τ_e is the pseudo-transient time constant. As previously emphasized, the dynamics have no physical significance and are only a mathematical device. The steady-state operation is solved when the pseudo-transient process model (5.4) is integrated to steady state. Additionally, initial conditions for the pseudo-transient temperatures must be provided. Fixing these at the inlet temperatures of their respective streams is a simple and reliable [192] option:

$$T_s^0(z) = T_{s,in} \tag{5.5}$$

Stability considerations for this reformulation are given in previous work [190, 192].

Remark 5.1. Consider the algebraic loop present in the steady state model of the (multi-stream) heat exchanger:

1. Stream enthalpies are calculated in (5.1) based on temperature driving force and the heat transfer coefficient
2. Stream temperatures are calculated implicitly from enthalpy
3. Heat transfer coefficients are calculated from stream temperatures (and phases)

Reformulating the model to make stream temperatures pseudo-transient state variables in (5.4) breaks this loop by enabling explicit calculation of the heat transfer coefficient and the temperature driving force at the initial conditions $T_s^0(z)$. Thus, the governing equations (5.1) and the flash calculations can be solved explicitly at the consistent initialization

step. Then (5.4) defines the evolution of the stream temperatures throughout the pseudo-time integration horizon. At steady state, the time derivative in (5.4) is zero, and the stream temperatures are at their steady-state values, satisfying (5.2) .

5.2.4 Pressure Drop Calculations

The detailed design of a MHEX is typically carried out after the full process is designed with an assumed fixed pressure drop in each stream that varies linearly with enthalpy [122, 192]. However, it may not be feasible to design a heat exchanger that meets the required specifications with the assumed pressure drops, especially when the streams change phase within the heat exchanger. Thus, when considering the detailed design of the SWHE simultaneously with the process design, I employ semi-empirical pressure drop correlations to ensure design feasibility and increase accuracy.

I assume that the heat exchanger is vertically-oriented and calculate stream pressure drops in the z dimension for each stream $s \in S$ as the sum of gravitational and frictional contributions. Note that the assumption of negligible pressure loss due to acceleration is valid at higher pressures, but could lead to slight inaccuracy at lower pressures [64]; however, the modeling framework presented is general, and the effect of kinetic energy on total pressure drop could be easily included in the model formulation for processes operating at lower pressures. The total pressure drop is thus approximated:

$$\frac{\partial P_s}{\partial z} = \left(\frac{\partial P_s}{\partial z}\right)_g + \left(\frac{\partial P_s}{\partial z}\right)_f \tag{5.6}$$

The pressure of stream s along the axial dimension (z) of the heat exchanger can then be calculated by numerical integration of (5.6):

$$P_s(z) = P_s^0 - \int_0^z \left(\frac{\partial P_s}{\partial z'}\right)_g + \left(\frac{\partial P_s}{\partial z'}\right)_f dz' \tag{5.7}$$

where P_s^0 is the stream inlet pressure. Note that for streams flowing countercurrent, the integration should be from the outlet location to location z rather than from the inlet location, 0, to z. Intuitively, the stream properties (phase, temperature, heat transfer coefficients, and even pressure itself) are dependent on the stream pressures. To simplify consistent initialization of the model, the continuation parameter $\hat{\alpha}$ is introduced as a multiplier to the pressure drop terms. Equation (5.7) is reformulated as:

$$P_s(z) = P_s^0 - \hat{\alpha} \int_0^z \left(\frac{\partial P_s}{\partial z'} \right)_g + \left(\frac{\partial P_s}{\partial z'} \right)_f dz' \tag{5.8}$$

The pseudo-transient evolution of the continuation parameter is described by (3.11)–(3.3.2), such that $\hat{\alpha}$ begins at zero and exponentially approaches $\hat{\alpha} = 1$ through the pseudo-time integration. This strategy effectively turns off the effect of pressure drop at the consistent initialization step, and allows for an explicit calculation of the stream properties from the stream inlet pressure along the length of the heat exchanger. As the pseudo-time integration proceeds, the effect of the pressure drop is gradually enforced, until it is fully present.

The gravitational pressure drop for stream s is given by:

$$\left(\frac{\partial P_s}{\partial z} \right)_g = g \rho_s(z) \tag{5.9}$$

where g is gravitational acceleration and ρ_s is the density of stream s. Empirical correlations from literature are used to compute frictional pressure drops and are described below.

Shell-Side Frictional Pressure Drop. For shell-side streams, the vapor contribution to frictional pressure drop for stream s is computed using a correlation provided by Neeras et al. [180]:

$$\left(\frac{\partial P_s}{\partial z}\right)_f = \mathcal{F}_v \frac{M_{s,v}^2}{2\rho_{s,v} p_l} \tag{5.10}$$

$$\mathcal{F}_v = \frac{2}{\mathcal{F}_{in}^{-0.5} + \mathcal{F}_{st}^{-0.5}} \tag{5.11}$$

$$\mathcal{F}_{in} = \mathcal{F}_{in,0}^2 \mathcal{F}_v^{-n} \tag{5.12}$$

$$\mathcal{F}_{st} = \mathcal{F}_{st,0}^2 \mathcal{F}_v^{-m} \tag{5.13}$$

where $M_{s,v}$ is the vapor mass flux of stream s, $\rho_{s,v}$ is the density of vapor in stream s, p_l is the longitudinal pitch of the heat exchanger tubes, and $\mathcal{F}_v$ is the friction factor for the vapor flow in stream s, $\mathcal{F}_{st,0}$ and $\mathcal{F}_{in,0}$ respectively represent the staggered and in-line contributions to the friction factor:

$$\mathcal{F}_{st,0} = 0.88\left[\frac{2a-1}{\sqrt{a^2+0.25b^2}} + 1\right]^2 \left[\frac{2(a-1)}{2a-1}\right]^{1.73} Re^m \tag{5.14}$$

$$\mathcal{F}_{in,0} = \begin{cases} 1.55(a-1)^{-0.7}(b-1)^{0.2} Re^n, & a \le b \\ 0.32\left[\frac{a-1}{b-1} - 0.9\right]^{-0.68}(b-1)^{-0.5} Re^n, & b > a \end{cases} \tag{5.15}$$

$$n = \begin{cases} 0.2, & a \le b \\ 0.2\left[\frac{b-1}{a-1}\right]^2, & b > a \end{cases} \tag{5.16}$$

$$a = \frac{p_r}{d_{tube}},\ b = \frac{p_r}{d_{tube}} \tag{5.17}$$

where p_r is the radial pitch of the heat exchanger tubes and m is a constant empirically determined to be 0.295 [180]. Re is the Reynolds number of the vapor flow:

$$Re = \frac{\pi d_{tube} M_{s,v}}{2\gamma\mu} \tag{5.18}$$

$$\gamma = 1 - \frac{\pi d_{tube}}{4 p_r} \tag{5.19}$$

At Re-numbers greater than 10^5, the friction factor is determined experimentally to be constant and given by the value calculated at $Re = 10^5$ [180]. The frictional pressure drop

from the liquid flow in the shell-side fluid is assumed to be negligible compared to the vapor-phase frictional pressure drop, so the frictional pressure in the two phase region is calculated considering only the contributions from vapor flow.

Tube-Side Frictional Pressure Drop. The frictional pressure drop for the tube-side streams s is computed using the correlation developed by Müller-Steinhagen and Heck [177]:

$$\left(\frac{\partial P_s}{\partial z}\right)_f = \left[\left(\frac{\partial P_s}{\partial z}\right)_{f,l} + 2X_s\left[\left(\frac{\partial P_s}{\partial z}\right)_{f,v} - \left(\frac{\partial P_s}{\partial z}\right)_{f,l}\right]\right](1-X_s)^{1/3} + \left(\frac{\partial P_s}{\partial z}\right)_{f,v} X_s^3 \quad (5.20)$$

where X_s is the flow quality of stream s (i.e., vapor mass fraction which is computed using a flash calculation) and $\left(\frac{\partial P_s}{\partial z}\right)_{f,l}$ and $\left(\frac{\partial P_s}{\partial z}\right)_{f,v}$ are respectively the frictional pressure drops of pure liquid and pure vapor in stream s:

$$\left(\frac{\partial P_s}{\partial z}\right)_{f,l} = \mathcal{F}_l \frac{M_s^2}{2\rho_l d_{tube}} \quad (5.21)$$

$$\left(\frac{\partial P_s}{\partial z}\right)_{f,v} = \mathcal{F}_v \frac{M_s^2}{2\rho_v d_{tube}} \quad (5.22)$$

$$F_p = \begin{cases} \frac{64}{Re_p}, & Re_p \leq 1187 \\ \frac{0.3164}{Re_p^{0.25}}, & Re_p > 1187 \end{cases} \quad (5.23)$$

where M_s represents the mass flux, $\mathcal{F}_p$ represents the friction factor of pure phase p, and Re_p represents the Reynolds number of pure phase p.

5.2.5 Heat Transfer Coefficients

Similar to pressure losses, heat transfer coefficients are highly dependent on the phase, composition, and temperature of each stream, as well as the heat exchanger geometry. Accounting for variations in the heat transfer coefficients allows for a more accurate calculation of the performance and required size (capital cost) of the heat exchanger.

Shell-Side Heat Transfer Coefficient. On the shell side, the liquid phase heat transfer coefficients, denoted as α_l, can be calculated using the empirical correlations provided by Neeras et al. [179]:

$$Nu_l = \frac{\alpha_l \delta_c}{\lambda} = a\left(\frac{d_{tube}}{\delta_c}\right)^c Re_l^b Pr_l^d \tag{5.24}$$

where λ is the fluid thermal conductivity, and δ_c is the reference film thickness:

$$\delta_c = \left[\frac{\mu^2}{g\rho^2}\right]^{1/3} \tag{5.25}$$

Pr_l is the Prandtl number Pr for the liquid phase:

$$Pr = \frac{C_p \mu}{k} \tag{5.26}$$

Re_l is liquid Reynolds number on the shell side, given by:

$$Re_l = \frac{M_{s,l}}{2\mu d_{total}} \tag{5.27}$$

$$d_{total} = \pi \frac{d_{outer} + d_{inner}}{2} N_{layers} \tag{5.28}$$

where $M_{s,l}$ is the shell side liquid mass flux and N_{layers} is the number of parallel radial tube layers (accounting for all tube-side streams) in the exchanger. The parameters a, b, c, and d were obtained from experimental data [180] and determined to be:

$$a = \begin{cases} 0.762, & Re_l \leq 2000 \\ 0.269, & Re_l > 2000 \end{cases} \tag{5.29}$$

$$b = \begin{cases} 1/9, & Re_l \leq 2000 \\ 1/4, & Re_l > 2000 \end{cases} \tag{5.30}$$

$$c = -1/3 \tag{5.31}$$

$$d = 1/3 \tag{5.32}$$

The vapor phase heat transfer coefficients α_v on the shell side can also be calculated using an empirical correlation given by Neeras et al. [180]:

$$Nu_v = \frac{\alpha_v \pi d_{tube}}{2\lambda} = f_A(0.3 + \sqrt{Nu_{lam}^2 + Nu_{turb}^2}) \tag{5.33}$$

where Nu_{lam} and Nu_{turb} are respectively the Nusselt numbers of the vapor phase for laminar and turbulent flow:

$$Nu_{lam} = 0.664 Re^{1/2} Pr^{1/3} \tag{5.34}$$

$$Nu_{turb} = \frac{0.037 Re^{4/5} Pr}{1 + 2.443 Re^{-0.1}(Pr^{2/3} - 1)} \tag{5.35}$$

The arrangement factor f_A in (5.33) is given empirically as:

$$f_A = 1 + \frac{0.4 + \frac{p_l}{p_r}}{\gamma^{3/2}(\frac{p_l}{p_r} + 0.7)^2} \tag{5.36}$$

where p_l and p_r are respectively the longitudinal and radial pitch in the exchanger (see Figure 5.2). γ is the void fraction used to calculate the average velocity between the tubes and is computed using (5.19).

Finally, the heat transfer coefficient on the shell side for two-phase flow is approximated as the linear sum of the vapor and liquid contributions to the heat transfer coefficient (each of the two contributions is calculated using only the portion of shell-side flow in the respective phase):

$$\alpha_{shell} = \alpha_l + \alpha_v \tag{5.37}$$

Tube-Side Heat Transfer Coefficient. On the tube side, the Boyko-Kruzhilin equation is used to estimate the heat transfer coefficient α_{tube} [36]:

$$Nu = \frac{\alpha_{tube} d_{tube}}{\lambda} = 0.024 Re_l^{0.8} Pr_l^{0.43} \sqrt{1 + X_s(\frac{\rho_l - \rho_v}{\rho_v})} \tag{5.38}$$

where Re_l and Pr_l are the liquid Reynolds number and Prandtl number on the tube side and X_s is the flow quality of stream s. ρ_l and ρ_v are respectively the liquid and vapor densities on the tube side, and λ is the thermal conductivity.

Overall Heat Transfer Coefficient. The heat transfer surface of each tube is elliptical, as shown in Figure 5.3.

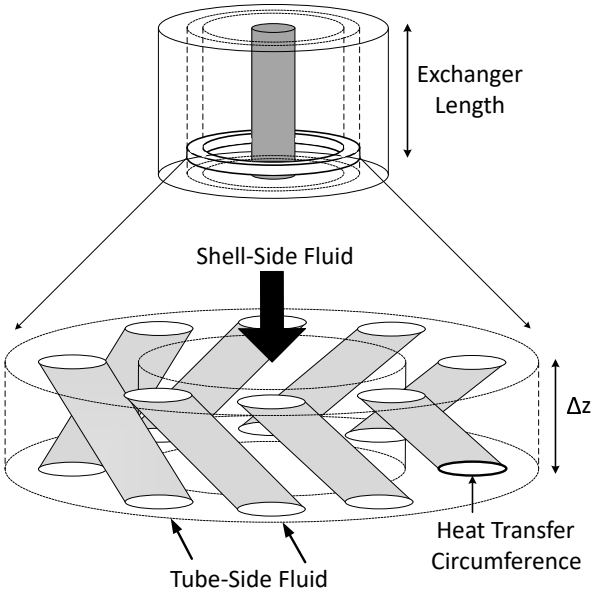

Figure 5.3: A simple depiction of a horizontal slice of a single radial layer of tubes in a spiral-wound heat exchanger.

The circumference of the ellipse with minor and major axes r_1 and r_2 is computed using Ramanujan's approximation [19]:

$$p \approx \pi\big[3(r_1 + r_2) - \sqrt{(3r_1 + r_2)(r_1 + 3r_2)}\ \big] \tag{5.39}$$

Considering the area $\bar{A}$ on a per-unit length basis, the overall differential heat transfer coefficient $U\bar{A}_{i,j}(z)$ between shell-side stream j and tube-side stream i is computed as [26]:

$$\frac{1}{U\bar{A}_{i,j}(z)} = \frac{1}{n_i p_{i,outer}\alpha_{shell}(z)} + \frac{1}{n_i p_{i,avg}\lambda_{tube} l_i^{-1}} + \frac{1}{n_i p_{i,inner}\alpha_{tube}(z)} \tag{5.40}$$

where $p_{i,outer}$, $p_{i,inner}$, and $p_{i,average}$ are respectively the outer, inner, and average heat transfer circumference for the tube of stream i, all computed using (5.39). The perimeters are multiplied by n_i the number of tubes containing stream i in a horizontal slice of the exchanger, to give the aggregate overall heat transfer coefficient between stream i and the shell-side fluid in a horizontal slice of the exchanger. α_{shell} and α_{tube} are respectively the heat transfer coefficients on the shell and tube sides, computed from (5.24), (5.33), and (5.38). λ_{tube} is the thermal conductivity of the metal tube, and l_i is the thickness of the tubes corresponding to stream i.

5.3 Case Study: PRICO Liquefaction Process

The PRICO process is a previously studied [63, 122, 149, 192] natural gas liquefaction process employing a MHEX for cooling and liquefying natural gas using a single refrigerant loop. The process flowsheet is shown in Figure 5.4. A natural gas stream (S1) enters the MHEX at 25°C and 55 bar and is liquefied and subcooled to -155°C. The (1 kmol/s) natural gas stream is composed of 89.7% methane, 5.5% ethane, 1.8% propane, 0.1% butane, and 2.8% nitrogen, and a mixed refrigerant composed of the same five components is used to cool the natural gas. The mixed refrigerant is employed in a single-stage refrigeration cycle, with the hot mixed refrigerant (HMR) being cooled in the tube side of the MHEX and expanded through a throttle valve. The cold mixed refrigerant (CMR) exiting the valve is then repassed through the shell side of the MHEX to liquefy the natural gas (which is also on the tube side), then compressed and chilled to 25°C in the salt water (SW) cooler. The compressor is assumed to operate with a fixed isentropic efficiency of 80%. The Soave Redlich Kwong cubic equation of state is used to model thermodynamic properties, and all fluid physical properties are calculated using mixing rules.

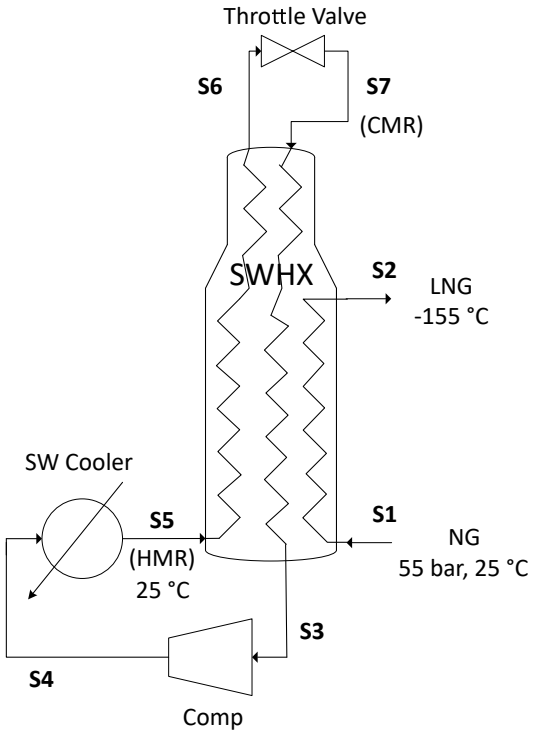

Figure 5.4: Process flow diagram for the PRICO liquefaction process.

The challenges arising during optimization of this process flowsheet include [192]:

- The phase of the refrigerant stream is unknown throughout the flowsheet.
- The pressures and temperatures of the refrigerant stream are variable throughout the flowsheet and within the SWHE.
- The composition and flow rate of refrigerant are decision variables for optimization, resulting in changing phase boundaries.
- A small minimum approach temperature is required to maximize energy efficiency.

Additionally, incorporating a detailed SWHE model into the process flowsheet during design optimization adds further challenges:

- The pressure drops of each stream through the heat exchanger are not fixed; pressure drops are calculated using semi-empirical correlations that are functions of temperature, composition, flow quality, and heat exchanger geometry (as described earlier).
- Rather than computing outlet temperatures based on satisfying the overall energy balance (as is the case with a conventional MHEX model used for process design calculations), the heat transfer is based on rate equations requiring that the heat transfer coefficients and heat exchange area also be calculated semi-empirically based on the local composition, temperature, pressure, flow quality, and heat exchanger geometry.
- The geometry of the heat exchanger (and thus pressure drops and heat transfer coefficients) changes between optimization iterations.

The cost of the PRICO process is dominated by the operating cost of the compressor, and optimization of the process often consists of minimizing the work done by the compressor [122, 63, 192]. The process operation decision variables comprise the refrigerant composition, refrigerant flow rate, and the high and low pressures in the refrigeration cycle. In addition to these, incorporation of a detailed heat exchanger model includes the geometrical aspects of the heat exchanger (Figure 5.2) as decision variables in optimizing the full process flowsheet. The SWHE design includes the exterior dimensions d_{outer}, d_{inner}, and L and the location of the side draw (at which point the liquefied natural gas stream S2 in Figure 5.4 exits the exchanger). The design of the heat exchanger also includes the angle β, longitudinal pitch p_l, and radial pitch p_r of both coils (natural gas and HMR—hot mixed refrigerant). The number of coils for both streams N_c can also be selected, as well as the diameter d_{tube} and thickness of the tubes l_{tube}. In this chapter, the number of coils N_c is treated as a continuous variable, which is a reasonable approximation due to the high number of coils for both hot streams. If higher accuracy is required, these variables can be treated as integer decisions

and the resulting optimal design problem could be solved using a MINLP (mixed integer nonlinear program) solver.

In addition to ensuring the liquefied natural gas product meets the required specification of being subcooled to -155°C, process constraints were included so that the pressure of the cold mixed refrigerant (CMR) at the end of the exchanger is above atmospheric pressure (1 bar) and that the CMR outlet temperature is above the dew point, such that the feed to the compressor is fully in the vapor phase. Geometrical constraints were also included in the optimization to ensure that the designed exchanger can be built in practice: (i) the space bars are at least 5 mm thick and (ii) the coils do not overlap radially or axially. Note that the detailed nature of the exchanger model permits excluding a constraint on minimum pinch temperature throughout the exchanger, as the heat exchange rate is calculated based on the temperature driving force (5.1).

5.3.1 Solution Details

Backwards finite differences are used to approximate the spatial derivatives of the hot streams in the axial domain of the heat exchanger and forward finite differences for the cold stream (where $z = 0$ is at the inlet of the hot streams and $z = L$ is the inlet of the cold stream). To initialize the design optimization, the initial set of optimization decision variables is selected as the midpoint of the respective bounds, and an optimization is first carried out to minimize the difference between the outlet natural gas stream temperature and the -155°C target. The result of this initial optimization is used as the initial guess for the design optimization (minimizing compressor power requirement). Although only operating cost, in the form of compressor power, is considered for this case study, capital cost for heat exchanger could easily also be estimated from the SWHE design specifications.

In total the system has 4327 equations, and the optimal solution found in 6888s of

CPU time using a 64-bit Windows 7 desktop system with a 3.40GHz Intel Core i7 processor and 16 GB of RAM. Of the total CPU time, 6629s (96.2%) were spent on state integration and ∼1s (0.009%) was spent by the optimization solver. The mathematical models were implemented and solved in gPROMS version 4.2.1 [204], and the sequential quadratic programming (NLPSQP) algorithm was selected to optimize the process. For all optimization procedures, the time relaxation-based approach for pseudo-transient models described in Chapter 3 was used [190].

5.3.2 Optimization Results

The list of process operation decision variables (i.e., pressures, temperatures, flow rates), their bounds, and their values at the optimal solution are shown in Table 5.1. Likewise, the heat exchanger geometric design decision variables, their bounds, and their values at the optimal solution are shown in Table 5.2. The natural gas tube coil is designated as NG and the hot mixed refrigerant coil is designated as HMR. It is important to note that the all the decision variables (Table 5.1 and Table 5.2) are optimized simultaneously and are only presented separately for clarity.

Table 5.1: PRICO process optimization results: Flowsheet level.

Decision Variable	Lower Bound	Optimal Value	Upper Bound
Refrig. N_2 (kmol/s)	0	0.305	5
Refrig. CH_4 (kmol/s)	0	1.052	5
Refrig. C_2H_6 (kmol/s)	0	1.376	5
Refrig. C_3H_8 (kmol/s)	0	0.059	5
Refrig. n-C_4H_6 (kmol/s)	0	0.901	5
Pressure S4 (bar)	10	25.93	50
Pressure S3 (bar)	1	4.26	10

The solution reveals that the optimal values for many of the decision variables are non-obvious design choices, as only a few decision variables reach their upper or lower bounds. As expected, the heat exchanger reaches its maximum allowed size—the heat exchanger

Table 5.2: PRICO process optimization results: Exchanger design.

Decision Variable	Lower Bound	Optimal Value (NG)	Optimal Value (HMR)	Upper Bound
Coil Angle (°)	3	3.76	3.34	15
Longitudinal Pitch (mm)	15	60.2	20.7	200
Radial Pitch (mm)	20	25	48.4	200
Space Bar Thickness (mm)	5	5	7.5	100
Number of Coils	4	22.6	23.6	35
Tube Diameter (mm)	20	20	43.4	200
Tube Thickness (mm)	1	1	1	5
SWHE Outer Diam. (m)*	1	5	5	5
SWHE Inner Diam. (m)*	1	1	1	5
SWHE Length (m)*	0	35	35	35
SWHE Side Draw (m)*	0	34.7	34.7	35

*The starred decision variables correspond to the full SWHE and are the same for both tube-side streams.

length at the optimal solution reaches its upper bound of 35 m, and the associated Lagrange multiplier at the optimum (shadow price) is -0.19 MW/m. The shadow price p_s is defined as

$$p_s = \frac{\partial J}{\partial x^b} \tag{5.41}$$

where J is the objective function and x^b is the value of the active constraint. Correspondingly, the local estimated effect of increasing the upper bound of the exchanger length by 1 m is a decrease in the work required (objective function) of 0.19 MW. At the optimal solution, the heat exchanger also reaches its maximum outer diameter and minimum inner diameter, and the tube thicknesses reach their minimum allowed values.

Interestingly, the natural gas stream passes through nearly the entire exchanger, and the ratio of cooling between the NG and HMR streams is controlled by the exchanger geometry, a consideration not posible when using less-detailed models. The hot and cold composite temperature-enthalpy curves at the optimum are shown in Figure 5.5. The optimized process shows tight heat integration, with a pinch point (ΔT_{min} = 1.35 K) at the cold refrigerant inlet of the heat exchanger.

Figure 5.6 shows the (nonlinear) stream pressure profiles along the heat exchanger.

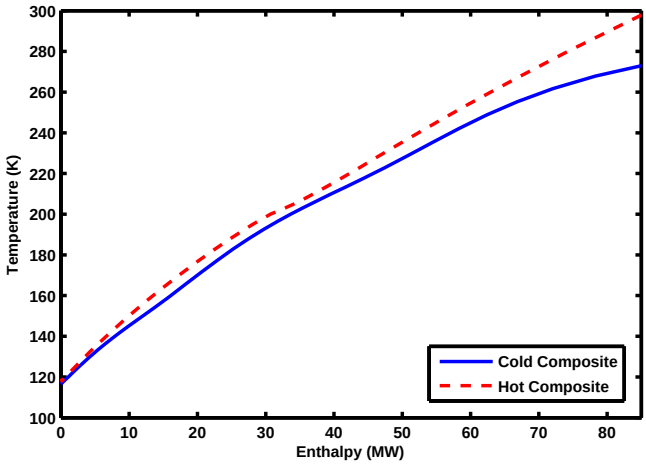

Figure 5.5: Composite curves for the optimized design of the PRICO process SWHE.

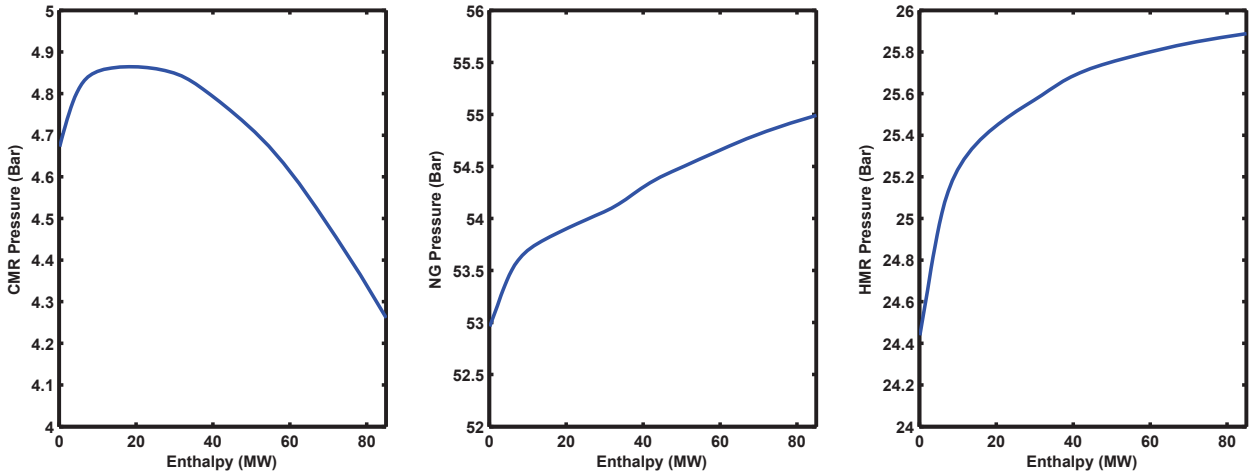

Figure 5.6: Stream pressures for the optimized PRICO liquefaction process SWHE. 0 MW represents the top of the heat exchanger, which is the inlet of the cold stream (shell side). Tube 1 contains the natural gas stream, while tube 2 contains the hot refrigerant.

In the shell-side stream, the pressure first increases when the stream is mostly liquid and the gravitational effects dominate the pressure drop as the fluid flows down the length of the exchanger. The pressure then begins to decrease as the stream vaporizes in the

exchanger and frictional effects begin to dominate the total pressure drop. Figure 5.7 shows that the heat transfer coefficients are also variable throughout the exchanger. Note that the presented modeling framework is general, and other semi-empirical heat transfer and pressure drop correlations than those proposed earlier can easily be incorporated into the model.

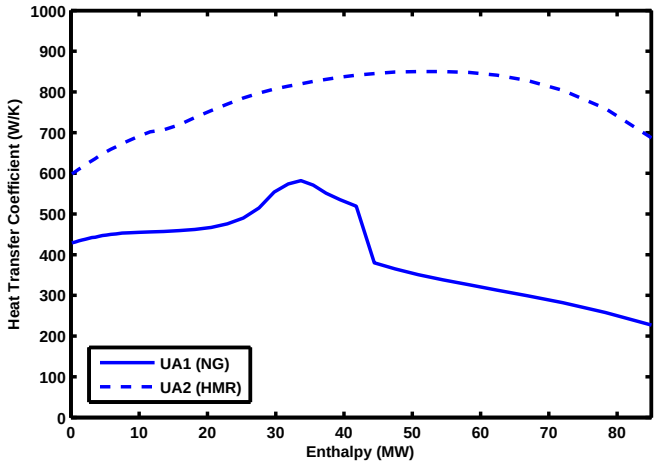

Figure 5.7: Heat transfer coefficients for the optimized PRICO liquefaction process SWHE. UA1 corresponds to the natural gas stream tubes, and UA2 corresponds to the hot refrigerant stream tubes.

Finally, the vapor flow quality of all heat exchanger streams is shown in Figure 5.8. The cold mixed refrigerant (CMR) stream is not fully vaporized until it reaches the outlet of the exchanger, indicating that the dew point temperature constraint is exactly met. The natural gas stream enters the exchanger in the vapor phase, undergoes phase change in a short two-phase region, and then is subcooled in the liquid phase. Both refrigerant streams (HMR and CMR) are predominately in the two-phase region along the exchanger.

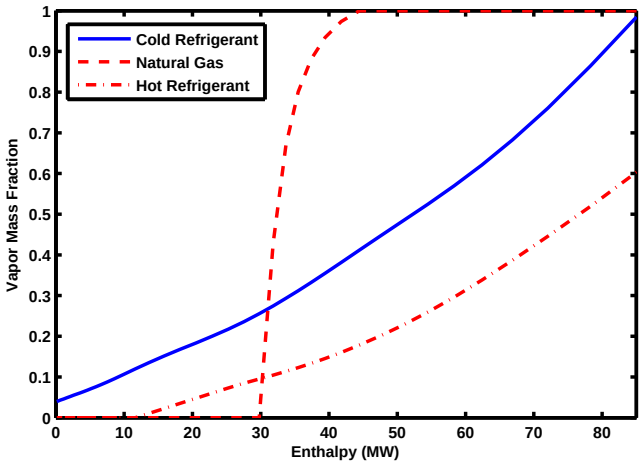

Figure 5.8: Stream flow qualities for the optimized PRICO liquefaction process SWHE. The inlet of the cold refrigerant stream is on the left, and the inlet of the hot refrigerant and the natural gas stream are on the right.

5.3.3 Discussion

The results of the optimization are presented alongside previous results from the literature in Table 5.3. Notably, although the optimal solution found in this work has a lower compression ratio than the solutions of previous works, it also uses more power than the best reported previous solution, due to a significantly increased refrigerant flow rate. The optimal solution of 20.34 MW is close to the previously reported optimum of 20.00 MW using a simplified thermodynamic heat exchanger model [192], with the shadow costs indicating that the design can be further improved by relaxing some constraints. The improvements over previous solutions stem from the adjustments in pressure drop and heat transfer allowed by simultaneously designing the SWHE and the flowsheet.

The large refrigerant flow rate reported in this work is a consequence of the spatially variable pressure drops calculated using the detailed model. In previous studies, it was assumed that the pressure drop across the shell side of the exchanger was fixed at 1 bar

Table 5.3: Comparison of PRICO process optimization results.

	Nogal et al. [63]	Kamath et al. [122]	Pattison & Baldea [192]	This Work
Power (MW)	24.53	21.51	20.00	20.34
Pressure S3 (bar)	4.84	2.02	3.38	4.26
Pressure S4 (bar)	43.87	17.13	26.55	25.93
Compression Ratio	9.1	8.5	7.9	6.1
Refrig. Flow (kmol/s)	3.53	2.93	2.89	3.69

[63, 122, 192] regardless of the stream properties (notably the phases present in the shell side). This study determined that, if the refrigerant stream vaporizes, the pressure drop increases dramatically, potentially leading to infeasible designs. A comparison between the temperature driving force, defined as the difference between the hot and cold composite temperatures, reported here and in the previous best result [192] is shown in Figure 5.9.

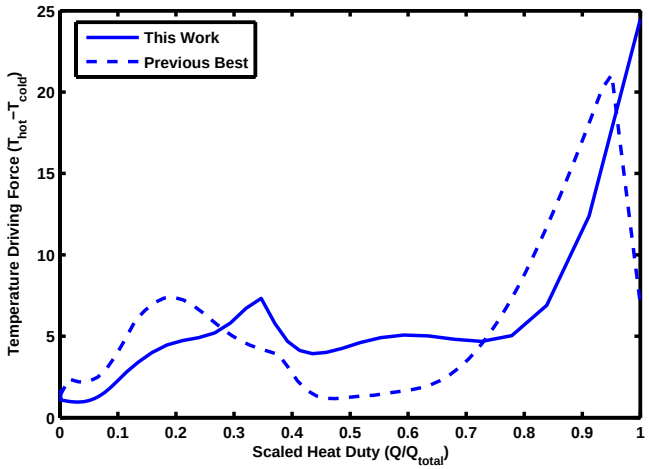

Figure 5.9: Temperature driving force along the optimized SWHE in this work and the previous best result [192].

Both exchanger designs include a small temperature driving force at the cold refrigerant inlet, but the previous result relies on a smaller refrigerant flow rate to achieve a smaller pinch point inside the exchanger (at approximately $Q/Q_{total} = 0.45$). In this previous design, the refrigerant completely vaporizes before exiting the heat exchanger, leading to lower

driving forces at the hot stream inlet, where the fully vaporized CMR stream is superheated. The solution found in this work does not involve superheating the CMR stream in the heat exchanger in order to maintain a low pressure drop, so the temperature driving force profile is maximized at the hot stream inlet (see the solid line in Figure 5.9). The compressor work is instead largely minimized by reducing the compression ratio—increasing the pressure at the outlet of the exchanger by lowering the pressure drop and decreasing the inlet pressure of the compressed HMR stream.

5.4 Summary

This chapter presented a novel approach for modeling, simulation, and optimization of spiral-wound heat exchangers (SWHEs) accounting for the geometric design of the coils and variations in stream properties along the axial dimension of the exchanger. The resulting pseudo-transient SWHE representation is easily combined with other process units models in a previously developed pseudo-transient process modeling, simulation, and optimization framework, allowing optimal design of a process and the associated heat exchangers to be carried out simultaneously. I demonstrated how the detailed SWHE model can be incorporated into process flowsheets in a natural gas liquefaction case study, finding that the results from previous studies are dramatically different from the result considering the heat exchanger geometry. The results highlight the importance of including detailed process unit models into flowsheet optimization when possible, such that critical equipment can be designed to match the performance specifications. I believe the methods used to model a SWHE in this chapter are general and can be applied to create models with different property correlations or for other types of multistream heat exchangers.

Chapter 6

Rate-Based Modeling and Optimization of Next-Generation Amine-Based Carbon Capture Plants[†]

In this chapter, I examine the simulation and optimization of advanced amine scrubbing processes with rate-based column models—another example of an important *complex, physical model*. I further show how the pseudo-transient strategies discussed in Chapter 3 are useful in regressing the model parameters using a detailed thermodynamic model derived from pilot plant data. Importantly, this step targets the necessity of *model validation and data reconciliation* capabilities in process flowsheets (see Section 2.6), which can allow process engineers to minimize model uncertainty during design optimization. Though the developments are general, I present an application to a plant utilizing a piperazine solvent, as well as a next-generation configuration of the absorber (with intercooling and pump-around) and stripping section with an advanced flash stripper. I consider two practically-relevant flue gas compositions (the unconventional "feedstock" to the process), showing a 14% annualized cost savings compared to a typical base-case process. The presentation in this chapter follows closely the material published in Tsay et al. (2019) [263].

[†]The contents of this chapter are largely based on the following publication: C. Tsay, R.C. Pattison, Y. Zhang, G.T. Rochelle, and M. Baldea. Rate-based modeling and economic optimization of next-generation amine-based carbon capture plants. Appl. Energy, 252:113379, 2019. C.T. is the primary author of the manuscript.

6.1 Carbon Capture Process Modeling and Optimization

Carbon emissions are a major contributor to global warming and climate change, and human-generated carbon emissions continue to increase with energy demand across the globe. As a result, carbon capture and sequestration, or CCS, remains a key technology for mitigating the effects of anthropogenic carbon emissions. Post-combustion CCS processes can greatly reduce the CO_2 emissions of existing coal and natural gas power plants. The Petra Nova facility, an absorption-based plant that captures 90% of the CO_2 from a flue gas stream, retained a million tons of CO_2 in just over six months of operation [75]. CO_2 captured by the Petra Nova project is utilized in enhanced oil recovery (EOR), one of many applications for captured CO_2. Such applications provide additional economic motivation for carbon capture, sequestration, and utilization (CCSU) with amine scrubbing. Additionally, new productive uses of carbon dioxide, such as microalgae cultivation and mineral carbonization, continue to emerge [205]. The broad applicability of CCS and CCSU, as well as increasing economic opportunities and incentives for utilization, suggest a long-term role for carbon-capture technologies in the future of sustainable energy.

Presently, the technology of choice for post-combustion CO_2 capture is amine scrubbing, which offers flexibility through relatively straightforward scale-up, potential for on/off operation dependent on demand, and retrofit possibilities for existing plants [38, 210]. However, amine scrubbing processes have high capital and operating costs, spurring efforts to improve process efficiency through advanced technologies, tight process integration, and process intensification [93, 109, 123, 173, 249]. Research efforts have also targeted the choice of absorbent [40]. While monoethanolamine (MEA) has been the standard solvent for amine scrubbing processes, piperazine (PZ) has been identified as a second-generation solvent because it is resistant to oxidative degradation, less volatile than MEA, and not corrosive to stainless steel [211, 298]. These advantages allow for better energy performance, but PZ-

based processes may exhibit precipitate and/or nitrosamine formation [38]. It is important to identify the optimal (e.g., in terms of economic efficiency) process configuration and operating conditions for PZ-based processes, which are likely different from those of a conventional MEA-based process [53, 155].

Experimental campaigns are important for identifying promising solvents and associated process configurations for CO_2 capture, but they are expensive and typically limited to small, pilot-scale systems [53, 173]. Consequently, mathematical modeling, simulation, and optimization can play a major role in identifying promising designs for carbon capture plants. Nevertheless, many commercially available models and software packages employ equilibrium-based predictions of the rate of CO_2 removal from flue gas (in the absorber) and rich absorbent (in the stripper). These equilibrium-based models can generally be reliably solved, but the performance of amine scrubbing processes is dominated by mass transfer kinetics, and rate-based (rather than equilibrium-based) models are needed to make accurate predictions [109, 295]. The solution of rate-based models presents many numerical difficulties for process simulation and optimization. As a result, previous studies often use reduced-order surrogate models to reproduce rate-based predictions and/or complicated thermodynamic models, sacrificing a degree of accuracy for ease of computation [60]. Advanced designs of amine scrubbing processes benefit from tight heat integration and multiple material and/or energy streams [93, 109], further complicating their simulation and optimization.

Literature Review. The literature contains numerous reports of prior research devoted to computed-aided modeling, simulation, and optimization of industrial-scale amine scrubbing plants for CO_2 capture. While this chapter addresses recently proposed advanced process configurations (i.e., absorber pump-around, intercooling, and advanced stripper configuration), most past efforts deviate minimally from the conventional process configuration shown in Figure 6.1.

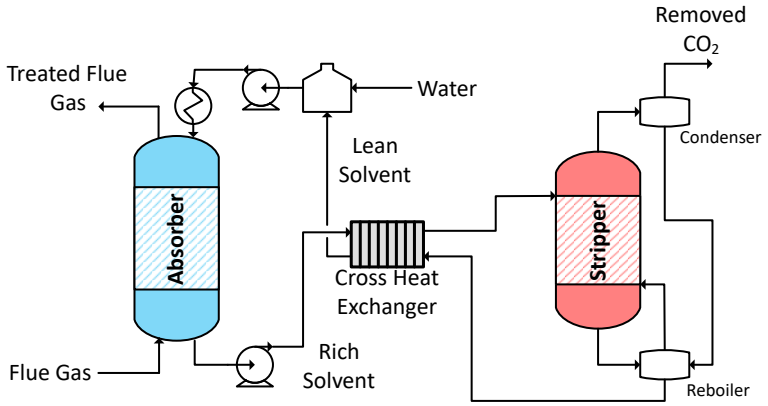

Figure 6.1: Conventional amine scrubbing process flowsheet

The conventional process (Figure 6.1) consists of a single absorber column, stripper column, and heat exchanger. The flue gas, from which CO_2 is to be removed, enters the bottom of the absorber column and contacts the down-flowing liquid solvent throughout the column. CO_2 is absorbed into the liquid solution through a reversible, exothermic reaction, and the treated flue gas is released to the atmosphere. The "rich" (having a high absorbed CO_2 concentration) solvent exits the bottom of the absorber and is sent to a stripping column. The stripping column operates at a higher temperature, reversing the absorption reaction. The CO_2 released from the stripper is compressed for sequestration or utilization, while the "lean" (having a low absorbed CO_2 concentration) solvent is recycled to the top of the absorber column. A cross heat exchanger recovers some heat from the lean solvent.

Improvements to the conventional amine scrubbing process have been identified via pilot-scale experiments [53, 93, 298] and *in silico* studies [60, 124, 187], with most works agreeing that the economically optimal process design is highly dependent on solvent choice and flue gas composition. It is notable that only a limited number of publications address optimization of the entire amine scrubbing process flowsheet, or the simultaneous design of unit operation, process configuration, and operating conditions that this chapter involves.

Among them, Mores et al. [174] used an equilibrium-based model to design a conventional amine scrubbing process including absorber size and operating conditions. Hasan et al. evaluated both the plant-scale [109] and multi-scale [110] economic impact of MEA-based absorption processes, using rate-based models and a decomposition-based procedure, whereby the process and heat integration structure are optimized separately. Burger et al. [40] incorporated solvent design into the process design optimization via a group-contribution method, with a reduced-order representation of the process and surrogate objective functions to aid in optimization. In a related reduced-order modeling effort, Nuchitprasittichai and Cremaschi [184] simultaneously optimized process design parameters, operating conditions, and solvent concentrations using response surface modeling and an artificial neural network as a surrogate for the objective function. More recently, Mores et al. [173] optimized a CO_2 capture process simultaneously with a coupled natural gas combined cycle plant. The aforementioned studies demonstrate the advantages of computer-based optimization of carbon-capture processes. However, they rely on many model simplifications and are limited in scope to the conventional process structure shown in Figure 6.1.

Many efforts have bee devoted to improving the efficiency of the energy-intensive stripping section. Alhajaj et al. [4] created a comprehensive economic evaluation method and employed equilibrium-based models to optimize and analyze the conventional amine scrubbing process. Optimization was performed by identifying the independent effect of individual operating parameters on various performance indicators (e.g., see Figure 2.3), ignoring multivariate interactions/combined effects. Damartzis et al. [60] used a superstructure-based approach to optimize flowsheet configurations for various solvents, using polynomial approximations of thermodynamic equilibrium. Karimi et al. [124] showed that carefully designed inter-heating in the stripping section can considerably decrease its overall energy requirement. Rochelle et al. [93, 53, 155] examined the energy performance of several alternative

stripper configurations, finding that energy efficiency is extremely sensitive to process operating conditions. In a more general effort, Oh et al. [187] used a superstructure to identify process improvements (e.g., absorber intercooling, stream splitting) with equilibrium-based models and a genetic algorithm optimization procedure.

As noted, many of the above works rely exclusively on equilibrium-based predictions. However, Zhang et al. [295] demonstrated the improved accuracy of rate-based models using extensive data from an MEA-based, pilot-scale carbon capture process. Several studies have employed such rate-based models: Zhang and Guo [291] validated a rate-based model and improved energy efficiency by examining the effects of a few key process parameters. Moioli and Pellegrini [172] proposed a rate-based model for PZ based on Eddy Diffusivity Theory and demonstrated its accuracy to experimental measurements. Also focusing on PZ, Zhang et al. [298] created a rate-based model with detailed thermodynamics and a novel stripper configuration, using pilot plant data to validate the model.

6.2 Modeling of Next-Generation Carbon Capture Process

This chapter considers a PZ-based carbon capture process flowsheet as described by Freeman et al. [94], shown in Figure 6.2. This process follows the general configuration of the conventional amine scrubbing process (Figure 6.1), but is significantly more complicated.

The process employs an advanced flash stripper—an interconnected flash tank and stripper column—with cold and warm bypass streams. The flue gas stream enters the bottom of the intercooled absorber (C1) at 40°C and 1 atm. *Rich* absorbent exits the bottom of the intercooled absorber (S1) and passes through two heat exchangers, HX1 and HX2, with a cold bypass stream (S3) removed before HX1 and a warm bypass stream (S5) before HX2. The remaining stream (S6) is heated and flashed in F1, with the liquid (S8) *lean* absorbent

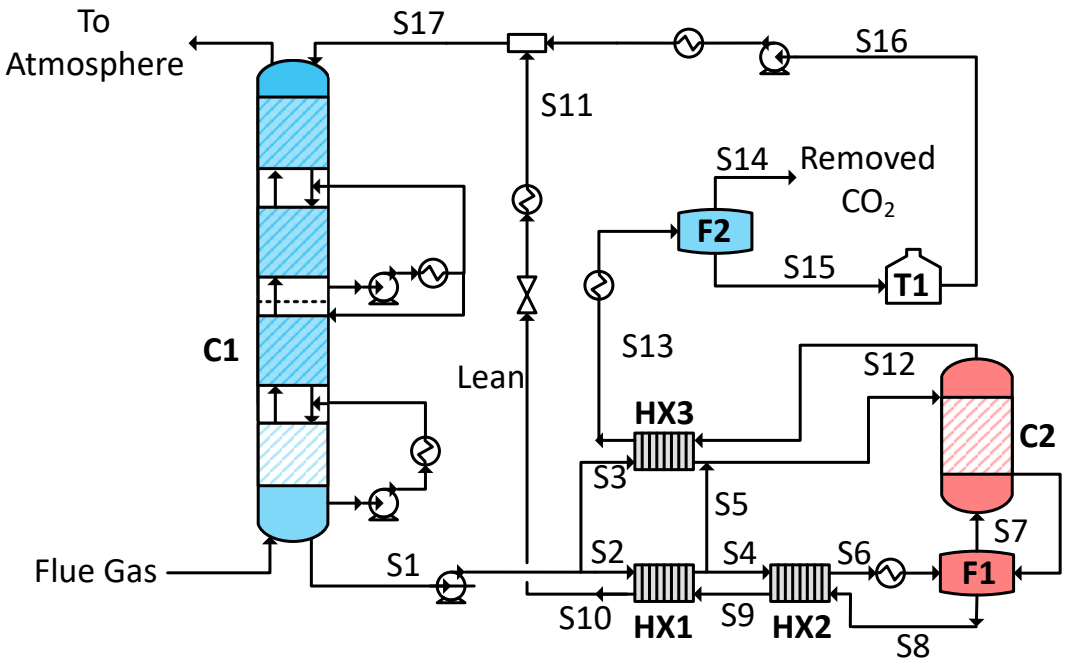

Figure 6.2: Advanced amine scrubbing process flowsheet.

passed back through HX1 and HX2. The vapor outlet of the flash (S7) passes through a stripper column (C2), after which the vapor (S12) is sent through a third heat exchanger, HX3, and is flashed at 40°C to separate CO_2 and water. The CO_2 (S14) is pressurized to 150 bar for sequestration or utilization, while the water (S15) is sent to a storage tank (T1) to be mixed with the lean absorbent. The cold bypass stream (S3) passes through HX3, is combined with S5, and enters the top of the stripper column (S15).

At intermediate points in the *intercooled* absorber, liquid is drawn out, cooled, and pumped back into the absorber. The location where liquid is pumped back can be above (pumparound) or the same as (in-and-out) the point where it is drawn from the column. As CO_2 is absorbed into the solvent in the absorber sections, the exothermic reaction causes the temperature of the down-coming solvent to increase, and water in the solvent vaporizes. The colder solvent toward the top of the absorber re-condenses water from the vapor phase,

and the two phenomena result in a spike in temperature within the absorber, referred to as a "temperature bulge." Intercooling an absorber reduces the magnitude of the temperature bulge by removing heat at intermediate points. The lower temperature improves the mass transfer rates throughout the column, especially when the bulge would otherwise occur towards the midpoint of the absorber height [296]. In-and-out cooling reduces the temperature bulge without increasing the liquid flow rate in sections of the absorber, while pumparound results in increased solvent flow in a particular section.

The stripping section of the process consists of three heat exchangers, a steam heater, a flash tank, and a stripper column. The split fractions $\xi_1 = \frac{S3}{S1}$ and $\xi = \frac{S5}{S2}$ (Figure 6.2) control the fraction of the cold and warm bypasses, which reflect the distribution of the rich absorbent between the flash and the top of the stripper column. ξ_1, ξ_2, and the size/duty of the heat exchangers are decision variables, and, consequently, the cold and hot bypasses can be removed at decision variable bounds. The heat exchangers can also be removed at these extreme cases, and the optimal point may not contain all three exchangers HX1, HX2, and HX3. The stripper column contains a sump tank at the bottom, where a liquid inventory is maintained for control purposes. While the advanced flash stripper is mathematically modeled as a flash tank and stripper column, in practice, *flashing*, or vapor-liquid separation, can occur in the stripper sump rather than in a separate process unit. Therefore, the flash tank is assumed to not contribute to the capital cost of the full process; rather, its cost is reflected in the cost of the stripper column.

6.2.1 Rate-Based Packed Column Model

For the purpose of modeling the carbon-capture plant, I develop a novel pseudo-transient, rate-based model for a packed column, which is used to represent the sections of the absorber and the stripper in the process flowsheet (Figure 6.2). The model is based on the

medium-order model proposed by Walters et al. [274], which considers variables distributed in the axial domain and represents the bulk liquid and vapor flows in the column using a plug-flow assumption. I further expand on this model to account for spatially dependent physical properties. The model considers the apparent species (H_2O, CO_2, and PZ), rather than capturing all true ionic species, present in the liquid phase. Mass and energy transport are calculated using two-film resistance theory with an apparent species mass transfer coefficient to describe the (unmodeled) boundary layer, as represented in Figure 6.3. PZ is assumed to be nonvolatile, and O_2 and N_2 in the flue gas are assumed to be insoluble. Bulk convection is assumed to be negligible, since the flue gas is therefore assumed to be mostly insoluble.

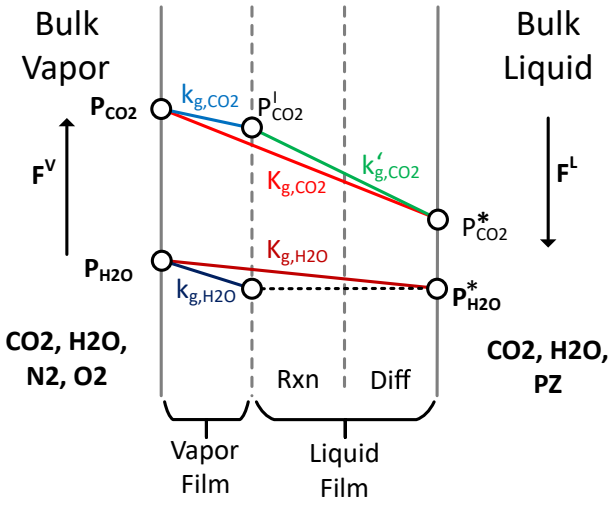

Figure 6.3: Two-film model for mass/energy transport. Mass transfer occurs at the boundary between bulk vapor and bulk liquid. In the gas film, there is a physical resistance to mass transfer $k_{g,i}$. In the liquid film, a rate-limited chemical reaction (Rxn) and diffusion (Diff) limit the mass transfer of CO_2. The two liquid resistances are lumped together as k'_{g,CO_2}. The total two-film resistance is denoted as $K_{g,i}$.

Material Balances and Interphase Flux. In each packed bed (or each absorber section), the axial domain is normalized, such that $z = 0$ at the bottom of the bed and $z = 1$ at the top. The model is non-dimensionalized by multiplying the spatial derivatives by the section height L. Figure 6.4 depicts the material balances for plug flow in a packed bed.

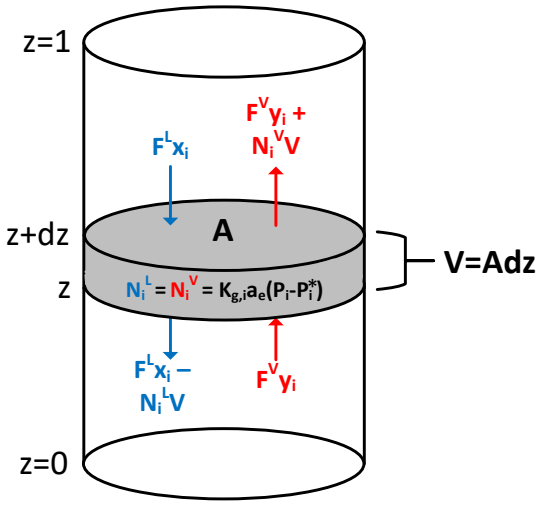

Figure 6.4: A schematic depiction of plug flow in a packed bed. The material flows of component i in and out of a horizontal "slice" of differential thickness dz are shown.

Assuming no radial variations, the plug-flow mass balance for each absorber section can be written as:

$$\frac{1}{LA}\frac{\partial(x_i F^L)}{\partial z} = N_i^L, \qquad i = \mathrm{CO_2}, \mathrm{H_2O}, \mathrm{PZ} \tag{6.1}$$

$$\frac{1}{LA}\frac{\partial(y_i F^V)}{\partial z} = N_i^V, \qquad i = \mathrm{CO_2}, \mathrm{H_2O}, \mathrm{N_2}, \mathrm{O_2} \tag{6.2}$$

$$\sum_i x_i = 1; \sum_i y_i = 1 \tag{6.3}$$

where F^L and F_V are respectively the liquid and vapor flow rates, x_i and y_i are the mole fractions, N_i^L, N_i^V represent the molar flux between the bulk liquid and bulk vapor, and $z \in [0, 1]$. A is the cross sectional area of the column. Equation (6.3) can be reformulated as a pseudo-transient model by defining the flow rates F^L and F^V as dynamic variables:

$$\tau_u \frac{\partial F^L}{\partial \hat{t}} = F^L(\sum_i x_i - 1) \tag{6.4}$$

$$\tau_u \frac{\partial F^V}{\partial \hat{t}} = F^V(\sum_i y_i - 1) \tag{6.5}$$

$$F^V(\hat{t} = 0) = F_{in}^V, F^L(\hat{t} = 0) = F_{in}^L \tag{6.6}$$

As shown in Figure 6.3, material transfer of CO_2 and H_2O occurs between the bulk liquid and bulk vapor within each section of column packing ($N_{PZ}^L = N_{N_2}^V = N_{O_2}^V = 0$). A linear driving force is assumed for each component flux to define the material transfer rate from bulk vapor to bulk liquid. Additionally, a continuation parameter $\hat{\alpha}$ is introduced to gradually introduce all material and energy fluxes, with dynamics described by (3.11)–(3.3.2).

$$N_i^L = N_i^V = \hat{\alpha}\big[K_{g,i}a_e(P_i - P_i^*)\big], i = \mathrm{CO_2}, \mathrm{H_2O}; z \in (0,1) \tag{6.7}$$

where $\hat{\alpha}$ is again a continuation parameter used to modulate the contribution of source/sink terms. P_i is the vapor pressure of component i calculated using Raoult's law, P_i^* is the equilibrium pressure of component i, and a_e is the effective area of packing. No mass flux is allowed at the ends of the column in order to ensure closure of the total material balance. The height of each column L is accordingly scaled using the height correction suggested by Walters et al. [273] to account for the loss of flux due to this boundary condition. The empirical correlation given by Xu [286] is used to approximate the equilibrium pressure of CO_2 (6.8), and the Antoine equation [96] is used to compute the equilibrium pressure of H_2O (6.9):

$$P_{CO_2}^* = \exp\big(35.3 - \frac{11054}{T^L} - 18.9\alpha^2 + \frac{4958}{T^L}\alpha + \frac{10163}{T^L}\alpha^2\big) \tag{6.8}$$

$$P_{H_2O}^* = x_{H_2O}10^{(8.07131 - \frac{1730.63}{T^L - 39.724})} \tag{6.9}$$

where T^L is the temperature of the bulk liquid and α is the CO_2 loading of the solvent. The *solvent loading* is defined as the ratio between the number of moles of CO_2 absorbed and the number of moles that can be carried by the solvent:

$$\alpha = \frac{x_{CO_2}}{2x_{PZ}} \tag{6.10}$$

Therefore, the rich absorbent has a high loading (close to 1), while the lean absorbent has a low loading (close to 0).

The mass transfer coefficient $K_{g,i}$ for H_2O is assumed to be equal to the gas film mass transfer coefficient for H_2O, while the mass transfer coefficient $K_{g,i}$ for CO_2 is determined through a series resistance model:

$$K_{g,H_2O}a_e = k_{g,H_2O}a_e \tag{6.11}$$

$$\frac{1}{K_{g,CO_2}a_e} = \frac{1}{k_{g,CO_2}a_e} + \frac{1}{k'_{g,CO_2}a_e} \tag{6.12}$$

where k_{g,H_2O} and k_{g,CO_2} respectively denote the gas film mass transfer coefficients of H_2O and CO_2. The effective mass-transfer area (a_e) depends on the type of contact surface, or *packing*, used in the column. k'_{g,CO_2} represents the mass transfer coefficient of the liquid phase, a combination of the kinetic resistance from a finite reaction and the physical resistance from the diffusion of reactants and products to and from the vapor/liquid interface. A power law expression fitted to experimental data is used to calculate gas film resistances for structured column packing [273], and a linear relationship is regressed to compute k'_{g,CO_2} as a function of loading. The effective area for liquid-side mass transfer is calculated as [252]:

$$k_{g,i}a_e = \left(\frac{1.07 \times 10^6}{RT^V}\right)(v^V)^{0.506}\left(\frac{D_{i,flue}}{10^4}\right)^{0.667}, i = CO_2, H_2O \tag{6.13}$$

$$\log(k'_{g,CO_2}) = c_1 - c_2\alpha \tag{6.14}$$

$$\frac{a_e}{a_p} = 1.34\left(\frac{\rho^L g^{1/3}|F^L|}{\sigma C^L L_p^{4/3}}\right)^{0.116} \tag{6.15}$$

where T^V is the temperature of the bulk vapor, and v^V is the velocity of the bulk vapor. $D_{i,flue}$ is the diffusion coefficient of component i in the flue gas, calculated using a mole-weighted average of binary diffusion coefficients computed using the Chapman-Enskog equation [33]. a_p is the nominal geometric area of the contact surface (packing), ρ^L is the liquid density, F^L is the liquid flow rate, σ is the surface tension, C^L is the liquid molar concentration, and L_p is the wetted perimeter of the packing. The rate constants c_1 and c_2 are fitted to data from a detailed, species-based thermodynamic model (Section 6.2.2).

Energy Balances and Heat Transfer. Assuming no radial variations, the energy balance for each absorber section can be written as:

$$\frac{1}{LA}\frac{\partial(h^L F^L)}{\partial z} = N_H^L \tag{6.16}$$

$$\frac{1}{LA}\frac{\partial(h^V F^V)}{\partial z} = N_H^V \tag{6.17}$$

where h^L is the specific molar enthalpy of the liquid flow, h^V is that of the vapor flow, and N_H^L and N_H^V represent the heat flux between the bulk liquid and bulk vapor. In order to instead compute temperature as a function of enthalpy, the temperatures of the bulk liquid and vapor phases, respectively denoted at T^L and T^V, are modeled using first-order pseudo-transient equations similar to (5.4):

$$\tau_{e,1}\frac{h_{in}^V}{T_0^V(z)}\frac{\partial T^V}{\partial \hat{t}} = h^V(z) - h^{PP,V}(T^V(z), P(z), \boldsymbol{y}(z)) \tag{6.18}$$

$$\tau_{e,2}\frac{h_{in}^L}{T_0^L(z)}\frac{\partial T^L}{\partial \hat{t}} = h^L(z) - h^{PP,L}(T^L(z), P(z), \boldsymbol{x}(z)) \tag{6.19}$$

where $h^{PP,L}$ and $h^{PP,V}$ denote the liquid and vapor enthalpies computed by a physical properties package using the appropriate equation(s) of state. The heat flux from the vapor phase to the liquid phase is calculated assuming only a gas film resistance due to the relatively high thermal conductivity of liquids. The gas/liquid interface temperature is assumed the same as that of the bulk liquid, and the rate of energy transfer can be written:

$$N_H^L = \hat{\alpha}\big[\sum_i N_i^L(h_i^V + \Delta H_{i,abs}) + \bar{h}a_e(T^V - T^L)\big] \tag{6.20}$$

$$N_H^V = \hat{\alpha}\big[\sum_i N_i^L(h_i^V) + \bar{h}a_e(T^V - T^L)\big] \tag{6.21}$$

where $\hat{\alpha}$ is the same continuation parameter for source/sink terms. The vapor enthalpy of each component is computed based on its heat capacity, and the heats of absorption for CO_2 and H_2O are calculated from equilibrium pressures (6.8)–(6.9) for thermodynamic

consistency:

$$h_i^V = C_p^V (T^V - T_{ref}) \tag{6.22}$$

$$\Delta H_{CO_2,abs} = -R(-11054 + 4958\alpha + 10163\alpha^2) \tag{6.23}$$

$$\Delta H_{H_2O,abs} = 1730.63R \frac{\log(10)}{(1 - 39.724T^L)^2} \tag{6.24}$$

The convective heat transfer coefficient $\bar{h}a_e$ is computed using the Chilton-Colburn analogy as a function of the gas film mass transfer coefficient for H_2O:

$$\bar{h}a_e = k_{g,H_2O}a_e \left(\frac{RT^V}{100}\right)(C^V C_p^V)^{1/3}\left(\frac{10^4\kappa^V}{D_{H_2O,flue}}\right)^{2/3} \tag{6.25}$$

where C_p^V and κ^V are respectively the heat capacity and thermal conductivity of the vapor phase.

6.2.2 Model Parameter Estimation

In this subsection, I discuss the regression of the parameters of the model described in Section 6.2.1. Unless otherwise noted, the physical properties of the flue gas and the vapor phase of the stripper column are computed assuming ideal gas behavior, and those of the absorbent are computed with the SAFT equation of state. The gSAFT [146] implementation was used with default parameters, as has been used in other recent works on carbon capture process simulation [40, 60, 213]. The general model reconciliation workflow involves simulating a model with rigorous thermodynamics and kinetics, collecting relevant data, and regressing the parameters of the model described here to fit the collected data.

An AspenPlus [9] model was previously developed by Frailie [92] with rigorous modeling of thermodynamics and kinetics, including liquid-phase speciation. The model has been fitted to experimental amine pK_a, CO_2 solubility, heat capacity, and amine volatility data for the PZ-H_2O-CO_2 system by regression of Gibbs free energy, enthalpy, heat capacity,

and activity coefficient parameters within the electrolyte non-random two liquid (e-NRTL) framework. Due to the aforementioned difficulties of commercial process simulators, the model uses separate simulations for the rate-based models of the absorber (modeled with 90 stages) and stripper (modeled with 35 stages). The two simulations are used in combination to simulate the full carbon capture process.

Intercooled Absorber. The intercooled absorber considered consists of four packed sections, each modeled as described in Section 6.2.1, and is shown in Figure 6.5. The bottom section (section 4 in Figure 6.5) consists of Mellapak 2X (MP2X) structured packing, while the remaining sections consist of Mellapak 250X (MP250X) structured packing. Other packing types can be modeled by substituting the appropriate geometric specifications [244].

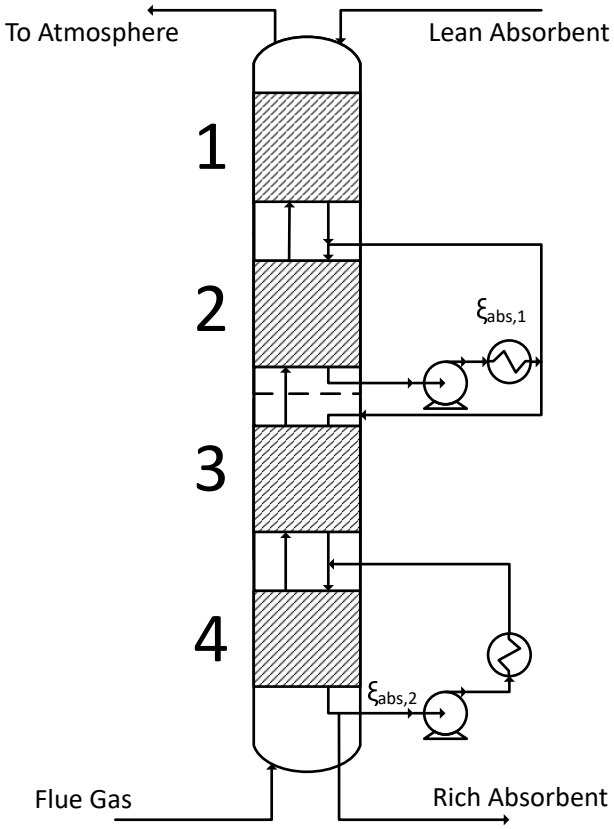

Figure 6.5: Flowsheet for absorber with four packed sections and intercooling

Two different absorber configurations based on the work of Freeman et al. [94] were simulated with the AspenPlus model: one designed for 90% CO_2 removal and the other for 99% removal. Both absorbers have a 15 m diameter. The parameters to describe k'_{g,CO_2} for the pseudo-transient model in (6.14) were regressed using data from the two different absorber simulations by minimizing the squared deviation from the rigorous model predictions:

$$\min_{c_1,c_2} \quad Error \tag{6.26}$$

$$Error = \int_z \frac{T^L - \hat{T}^L}{\hat{T}^L} + \int_z \frac{T^V - \hat{T}^V}{\hat{T}^V} + \sum_i \int_z \frac{F_i - \hat{F}_i}{\hat{F}_i} \tag{6.27}$$

The value for c_2 in (6.14) was enforced to be the same throughout the column, as it is related to the solvent kinetics and should not change with the packing type. The value for c_1 is related to the liquid diffusive resistance of the packing [252] and was allowed to differ between the MP250X and MP2X sections. The optimal values found were $c_1 = -4.96$, $c_2 = 13.34$ for the MP250X packing and $c_1 = -5.60$, $c_2 = 13.34$ for the MP2X packing. The agreement between the temperature and vapor concentration profiles of the pseudo-transient model and the rigorous AspenPlus model is generally very good, and a comparison between the fitted model and the training data is shown in Figure 6.6.

Furthermore, the same simulations were repeated with a smaller diameter of 13.5 m (packed section heights were adjusted to achieve 90% and 99% removal) to test the prediction accuracy of the regressed model, and the results are shown in Figure 6.7. The temperature and vapor concentration profiles predicted by the two models are very close, even on the generated test data, suggesting that the developed model can make accurate predictions for a range of absorber diameters and heights.

Stripper Column. The stripper column consists of a single packed section, filled with Raschig SuperRing No. 0.5 (RSR#0.5) random packing [227]. The stripper mathematical model has the same structure as that of the absorber, but differs in some important

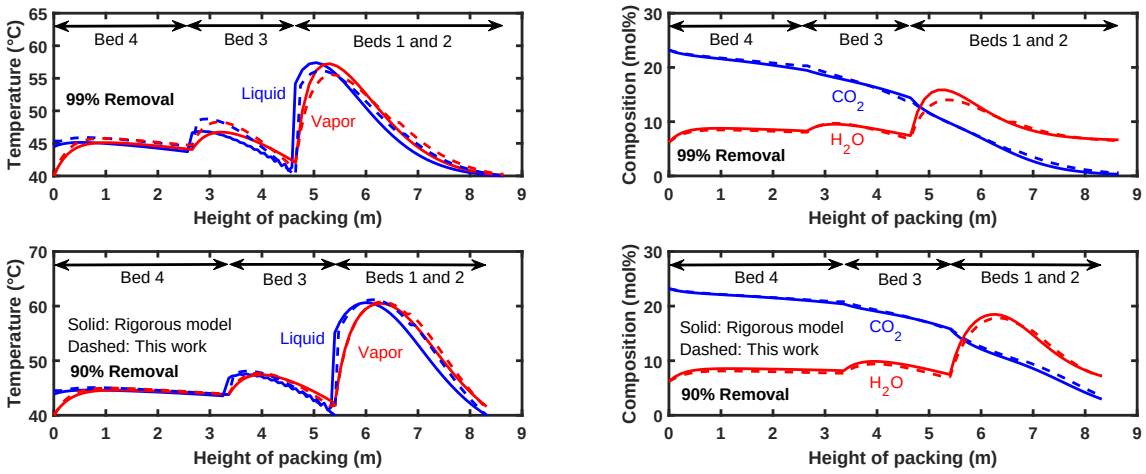

Figure 6.6: Absorber temperature profiles and gas concentrations for simulations used to train k'_{g,CO_2} parameters. The solid lines correspond to the values predicted by the rigorous AspenPlus model, which were used to fit the model. The dashed lines correspond to predictions from the fitted model.

details. First, the effective area of the random packing is calculated differently from that of structured packing, and (6.15) is replaced with an appropriate correlation for random packing [275]:

$$\frac{a_e}{a_p} = 1.41\left(\frac{\rho^L g^{1/3} v^{L\,4/3}}{\sigma a_p^{4/3}}\right)^{0.116} \tag{6.28}$$

Second, the expression for k_g given in (6.13) is scaled by a factor cs_1 for random packing, and new values of k'_g are regressed for the stripper. The mass transfer parameters are expected to be different in the stripper, as bulk convection may occur (the gases present are soluble) and mass transfer occurs in the reverse direction (i.e., CO_2 flows from vapor to liquid in the absorber and from liquid to vapor in the stripper).

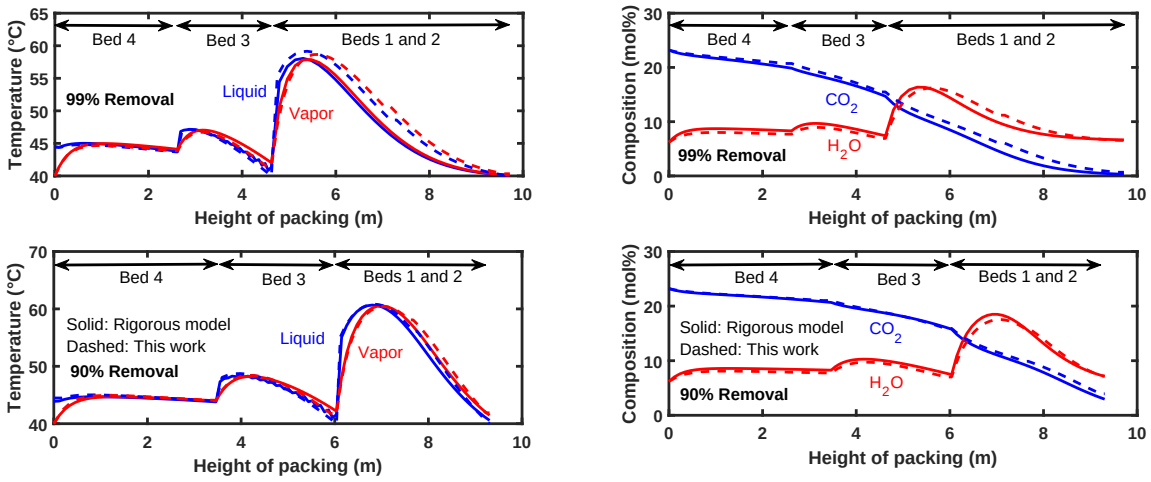

Figure 6.7: Absorber temperature profiles and gas concentrations for simulations used to test the fitted k'_{g,CO_2} parameters. The solid lines correspond to the values predicted by the rigorous AspenPlus model, using different heights and diameters from the training data in Figure 6.6. The dashed lines correspond to predictions from the fitted model.

Oxygen and nitrogen are absent in the stripper, and the gas film mass transfer coefficients of H_2O and CO_2, as calculated by (6.13), are equal. k_g and k'_g are thus computed:

$$k_{g,i}a_e = cs_1\left(\frac{1.07\times10^6}{RT^V}\right)(v^V)^{0.506}\left(\frac{D_{i,flue}}{10^4}\right)^{0.667}, i = \mathrm{CO_2}, \mathrm{H_2O} \tag{6.29}$$

$$\log(k'_{g,CO_2}) = cs'_1 - cs'_2\alpha \tag{6.30}$$

Two different stripper configurations were simulated with the rigorous AspenPlus model based on the 90% CO_2 removal and 99% removal absorber designs. Both were simulated assuming a 15 m column diameter [94]. The parameters to describe k'_{g,CO_2} were similarly regressed using data from the two stripper simulations by minimizing the squared deviation from the high-order model predictions. The optimal values found were $cs_1 = 0.25$, $cs'_1 = -0.03$, and $cs'_2 = 34.19$ for the RSR#0.5 packing. The agreement between the temperature and vapor concentration profiles between the regressed model and rigorous Aspen Plus simulations is not as good as in the case of the absorber model. The inaccuracy can possibly

be attributed to the contribution of bulk convection to mass transfer, since the vapor in the stripper is composed of soluble components. Further, the correlations employed are developed based on absorber data, since fewer previous works in experimental and modeling have been devoted to the modeling of stripper behavior. Nevertheless, the stripper comprises a much smaller portion of the capital cost than the absorber, and inaccuracies in the model are less likely to impact the optimal process design. A comparison between the fitted model and the training data is shown in Figure 6.8.

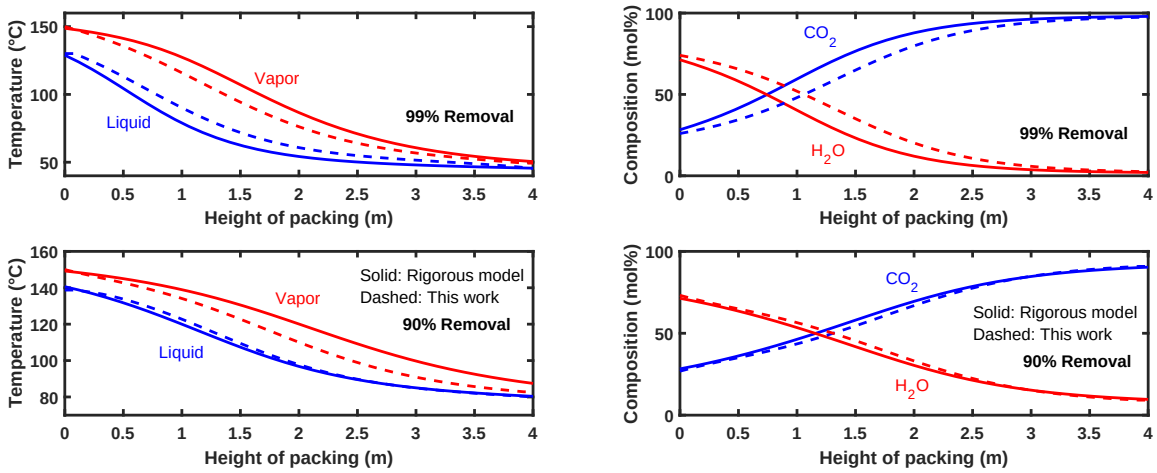

Figure 6.8: Stripper temperature profiles and gas concentrations for simulations used to train $k_{g,i}$ and k'_{g,CO_2} parameters. The solid lines correspond to the values predicted by the rigorous AspenPlus model, while the dashed lines correspond to the fitted model.

6.2.3 Modeling of Other Process Units

In addition to the absorber and stripper, the process flowsheet (Figure 6.2) contains two flash tanks (F1, F2) and three heat exchangers (HX1, HX2, HX3). Pseudo-transient models for flash tanks and heat exchangers have been previously developed [190] as part of a library of common unit operations, and the models are used here with a few minor modifications described here. The vapor fractions in the flash tank F1 were calculated with

the methodology described by Madan et al. [162] for PZ-CO_2-H_2O systems at high pressure and temperature:

$$\log(P_{CO_2}) = 35.5 - 11054\frac{1}{T} - 18.9\alpha^2 + 4958\frac{\alpha}{T} + 10163\frac{\alpha^2}{T} \tag{6.31}$$

$$\log(P_{H_2O}) = 73.6 - 7258\frac{1}{T} - 7.3\log(T) + 4.2 \times 10^{-6}T^2 \tag{6.32}$$

The second flash tank F2 separates only CO_2 and H_2O, and vapor-split fractions were computed with gSAFT. The model for heat exchangers HX1, HX2, and HX3 is given by equations (6.33)–(6.38). The temperatures of the outlet streams are calculated using first-order differential equations, as proposed by Pattison and Baldea [190]. The Underwood approximation [265] was used to define the driving force for heat transfer in (6.35), as it has better numerical properties than the log mean temperature difference [251].

$$F^H_{out}h^H_{out} = F^H_{in}h^H_{in} - Q \tag{6.33}$$

$$F^C_{out}h^C_{out} = F^C_{in}h^C_{in} - Q \tag{6.34}$$

$$Q = UA_{HX}\left(\frac{(T^H_{out} - T^C_{in})^{1/3} + (T^H_{in} - T^C_{out})^{1/3}}{2}\right)^3 \tag{6.35}$$

$$\tau_{e,2}\frac{h^H_{in}}{T^H_0}\frac{dT^H_{out}}{d\hat{t}} = h^H_{out} - h^{PP}(T^H_{out}, P^H_{out}, \boldsymbol{z}^H) \tag{6.36}$$

$$\tau_{e,2}\frac{h^C_{in}}{T^C_0}\frac{dT^C_{out}}{d\hat{t}} = h^C_{out} - h^{PP}(T^C_{out}, P^C_{out}, \boldsymbol{z}^C) \tag{6.37}$$

$$T^H_{out}(\hat{t}=0) = T^{0,H}, T^C_{out}(\hat{t}=0) = T^{0,C} \tag{6.38}$$

where the superscripts H and C refer to the hot and cold streams respectively. Note that (6.35) is used since log mean temperature difference $(\Delta T_1 - \Delta T_2)/\log(\Delta T_1/\Delta T_2)$ is indeterminate if $\Delta T_1 = \Delta T_2$ at any point during the pseudo-time integration. The overall heat transfer coefficients U of all three heat exchangers were assumed to be constant at 800 $W/m^2/K$, a value that is similar to that found in other works [173, 267]. The same value of U was used for all three heat exchangers to avoid optimal design solutions that divert flow (via cold and warm bypasses) to exchangers with higher overall heat transfer coefficients.

6.3 Optimization of Next-Generation Carbon Capture Process

The cost of the carbon capture plant has two components: capital cost (CAPEX) and operating cost (OPEX). The main contributors to CAPEX are the absorber, stripper, heat exchanger(s), steam heater, and compressor. The cost of the flash tank F1 was not included, since the flash is typically not a separate process unit as explained earlier. The Tsai method [252] was used to calculate the purchased equipment cost (PEC) of the absorber and stripper. The method is based on raw materials costs, and the estimate was scaled by a factor of five to represent PEC. The PEC of heat exchangers was calculated by assuming a fixed price of \$22.16 per square foot. Linear scaling (as opposed to a power law) is reasonable, as heat exchanger area is typically increased by adding heat exchangers in parallel for a plant of the capacity considered here. Economic correlations based on a survey of vendor prices [93, 155] were used to estimate the PEC of the steam heater and the compressor.

The OPEX consists of heating, cooling, and electricity costs. Heating costs were calculated by assuming a steam cost of 0.2 cents per MJ, cooling costs were assumed to be 0.063 cents per MJ, and the cost of electricity was assumed to be \$100 per MWh. I note that while the heating costs were assumed to be constant, in reality they may be related to the opportunity cost of electricity production for an integrated power plant. The total annualized capital cost can be calculated from PEC using expression (6.39) [155]. The scaling factor a converts PEC to total capital (installed) cost, while the factor b annualizes the cost. Direct cost, indirect cost, and working capital are included in a, and return on investment, taxes, depreciation, and maintenance are include in b. The values recommended by Lin and Rochelle [155] of $a = 5$ and $b = 0.2$ were used.

$$\text{Total annualized capital cost}\left(\frac{\$}{yr}\right) = a \times b \times \text{PEC}(\$) \tag{6.39}$$

The decision variables for the design of the intercooled absorber are the diameter of

the absorber, the heights of packed sections 1–4, and the split fractions $\xi_{abs,1}$ and $\xi_{abs,2}$ (see Figure 6.5). For the design of the stripping section, the decision variables are the diameter of the stripper, the height of the stripper, and the temperature and pressure of the flash tank F1. Finally, the process-level decision variables include the flow rate of solvent, the sizes of the heat exchangers (HX1, HX2, and HX3), the cold bypass fraction $\xi_1 = S3/S2$, and the warm bypass fraction $\xi_2 = S5/S4$. I emphasize that all decision variables, including unit-level and flowsheet-level decisions, were optimized simultaneously.

The solvent was assumed to have a fixed amine concentration of 5 molal piperazine. Although a higher concentration of piperazine may increase the solvent capacity, pilot plant studies [298] have shown that 5 molal piperazine outperforms higher concentration solvents due to lower solvent viscosity (enhanced diffusion and turbulence). In carrying out the design optimization calculations, the following constraints were enforced:

- The CO_2 removal must be at least 90%, i.e., $Removal = \frac{F^V_{CO_2,in} - F^V_{CO_2,out}}{F^V_{CO_2,in}} \geq 0.9$.
- The heat exchangers must have a minimum temperature difference (minimum approach) of at least 1°C, i.e., $\Delta T_{min} = \min(T^H_{out} - T^C_{in}, T^H_{in} - T^C_{out}) \geq 1°C$.
- The flooding ratios of both the absorber and stripper must be 70% or lower, i.e., $\frac{\max(F^V)}{F^V_{flood}} \leq 0.7$, where F^V_{flood} is the flooding point vapor flow rate.
- The pumparound ratios must be 50% or lower, since pressure losses in the absorber were excluded, i.e., $\xi_{abs,1}, \xi_{abs,2} \leq 50\%$.
- The temperature of the flash tank must not exceed 150°C in order to avoid thermal degradation of the solvent, i.e., $T_{F1} \leq 150°C$.
- The total bypass ratio of the plant must not exceed 50% to be consistent with pilot plant results [154], i.e., $\frac{S3+S5}{S2} \leq 0.5$.

Column flooding refers to excessive vapor flow that causes liquid to be entrained and carried upward. Close to the flooding point (defined as $F^V \approx F^V_{flood}$), the vapor velocity is high enough to carry an excessive amount of liquid upwards. This form of backmixing causes the column performance to decrease, and it also results in a sharp increase in column pressure drop. The flooding ratio $\frac{\max(F^V)}{F^V_{flood}}$ can be decreased by increasing the column diameter, which increases F^V_{flood} by dispersing vapor flow over a larger cross-sectional area (at the cost of a larger, more expensive column).

In carring out design optimization calculations, I first considered the case where a membrane system is used in conjunction with the amine scrubbing process (see [38, 94, 249] for a discussion and overview of membrane and hybrid-membrane systems). The hybrid configuration benefits from a higher inlet CO_2 concentration and potentially operates at a higher lean-loading state. I began with this case since a carbon capture plant design heuristically optimized for 90% removal is available as a reference point [94]. The reference base case process, shown in Figure 6.9 has a 12.9 kmol/s inlet flue gas flow rate, with the stream containing 23.2 mol% CO_2, 6.2 mol% H_2O, 69 mol% Nitrogen, and 1.6 mol% Oxygen. Finite differences in the reverse direction of flow was used to approximate the spatial partial derivatives in the absorber and stripper. Each section of the absorber was discretized using 20 finite difference points (80 total points), while the stripper was discretized using 50 finite difference points.

The resulting model has 10068 equations and was implemented in gPROMS version 5.1.4 [204], using the built-in DASOLV package for time integration. The values of the time constants $\boldsymbol{\tau}$ are selected to create a hierarchy of decoupled pseudo-transient dynamics (see Figure 3.2): $\tau_u = 10^0$ in (6.4)–(6.5), $\tau_{e,1} = 10^1$ in (6.18), and $\tau_{e,2} = 10^3$ in (6.19), (6.36)–(6.37). The value of the time constant for the continuation parameter is set to $\tau_\alpha = 10^{-1}$ in (3.11), and the dynamic tearing time constant is set to $\tau_r = 10^4$ in (3.12). Simulation of

the base case required 272s of CPU time using a 64-bit Windows 10 desktop system with a 3.20GHz Intel Core i7 processor and 16GB of RAM. The values of the decision variables in the base case are presented in Table 6.2, and the economic costs are presented in Table 6.1. The base case design removes 90% of CO_2 from the flue gas stream, capturing 4.12 million tons of CO_2 per year ($28.40 per ton of CO_2). The absorber operates at 67% flooding, while the stripper operates at 30% flooding. The minimum temperature approaches of HX1, HX2, and HX3 are respectively 15.00°C, 15.00°C, and 36.51°C.

6.3.1 Case 1: Base Case Optimization

The process was optimized using the pseudo-transient approach (Chapter 3) with the sequential quadratic programming algorithm (NLPSQP) in gPROMS [204]. The base case was used as the initial guess, and the optimal solution was found in 5551s of CPU time (58 NLP iterations). While the CPU time required for this study is large, recent results suggest that the pseudo-transient state integration time ($>$ 99.9% of CPU time) can be reduced by orders of magnitude by attempting solution of the original steady-state algebraic model during optimization iterations [158].

The optimal values of the decision variables and economics of the optimal design are shown in Table 6.2 and Table 6.1.

	Base Case		Optimal Point	
Capital Cost (PEC)	\$Million	contribution	\$Million	contribution
Absorber	18.11	15.5%	14.23	14.1%
Stripper	7.96	6.8%	5.17	5.1%
Exchangers	12.21	10.4%	12.53	12.4%
HX1	*5.60*		*3.38*	
HX2	*6.55*		*8.96*	
HX3	*0.06*		*0.19*	
Compressor	13.92	11.9%	13.93	13.8%
Steam Heater	2.08	1.8%	1.22	1.2%
Operating Cost	\$Million/yr	contribution	\$Million/yr	contribution
Heating	32.27	27.6%	22.32	22.1%
Cooling	6.37	5.4%	7.28	7.2%
Electricity	24.11	20.6%	24.14	23.9%
Totals	\$Million/yr	contribution	\$Million/yr	contribution
CAPEX	54.28	46.4%	47.07	46.7%
OPEX	62.76	53.6%	53.74	53.3%
TOTAL	**117.03**	**100.0%**	**100.82**	**100.0%**

Table 6.1: A comparison of process economics between base-case and optimal-point values for the case of 90% CO_2 removal from a rich flue gas stream.

	Lower Bound	Base Case	Optimal Point	Upper Bound
Intercooled Absorber				
Absorber Diameter (m)	0	15.00	13.59	-
Section 1 Height (m)	0	1.46	2.15	-
Section 2 Height (m)	0	1.46	1.59	-
Section 3 Height (m)	0	2.30	3.56	-
Section 4 Height (m)	0	3.36	0.00	-
Split 1 $\eta_{abs,1}$ (%)	0	0.0	32.0	50
Split 2 $\eta_{abs,2}$ (%)	0	50.0	0.0	50
Stripping Section				
Stripper Diameter (m)	0	13.05	8.74	-
Stripper Height (m)	0	4.00	4.92	-
Flash Temperature (°C)	0	150	150	150
Flash Pressure (bar)	1.01325	5.80	5.80	12
Cold Bypass η_1 (%)	0	6.0	4.0	50
Warm Bypass η_2 (%)	0	35.0	40.0	50
Other Process Decisions				
Solvent (S17) Flow (kmol/s)	0	89.12	93.53	-
HX1 Area (m^2)	0	23472	14157	-
HX2 Area (m^2)	0	27453	37555	-
HX3 Area (m^2)	0	260	802	-

Table 6.2: A comparison between base-case and optimal-point values for the process design variables for the case of 90% CO_2 removal from a rich flue gas stream.

The optimal flowsheet is shown in Figure 6.9. At the optimal point, 90% of CO_2 from the flue gas stream is removed, representing a capture of 4.12 million tons of CO_2 per year ($24.47 per ton of CO_2). The annual cost of the process and the cost per ton of CO_2 are both reduced by 14% compared to the base case. The absorber and stripper both operate at 70% flooding ratio, and their diameters are slightly reduced. The minimum approach temperatures of HX1, HX2, and HX3 are respectively 21.79°C, 8.16°C, and 9.25°C. I believe these design changes are non-obvious, with computer-guided optimization procedure increasing the minimum approach temperature of HX1 and decreasing those of HX2 and HX3. Correspondingly, the size (and contribution to capital cost) of HX1 decreases from the base case to the optimal point, while the sizes of HX2 and HX3 increase.

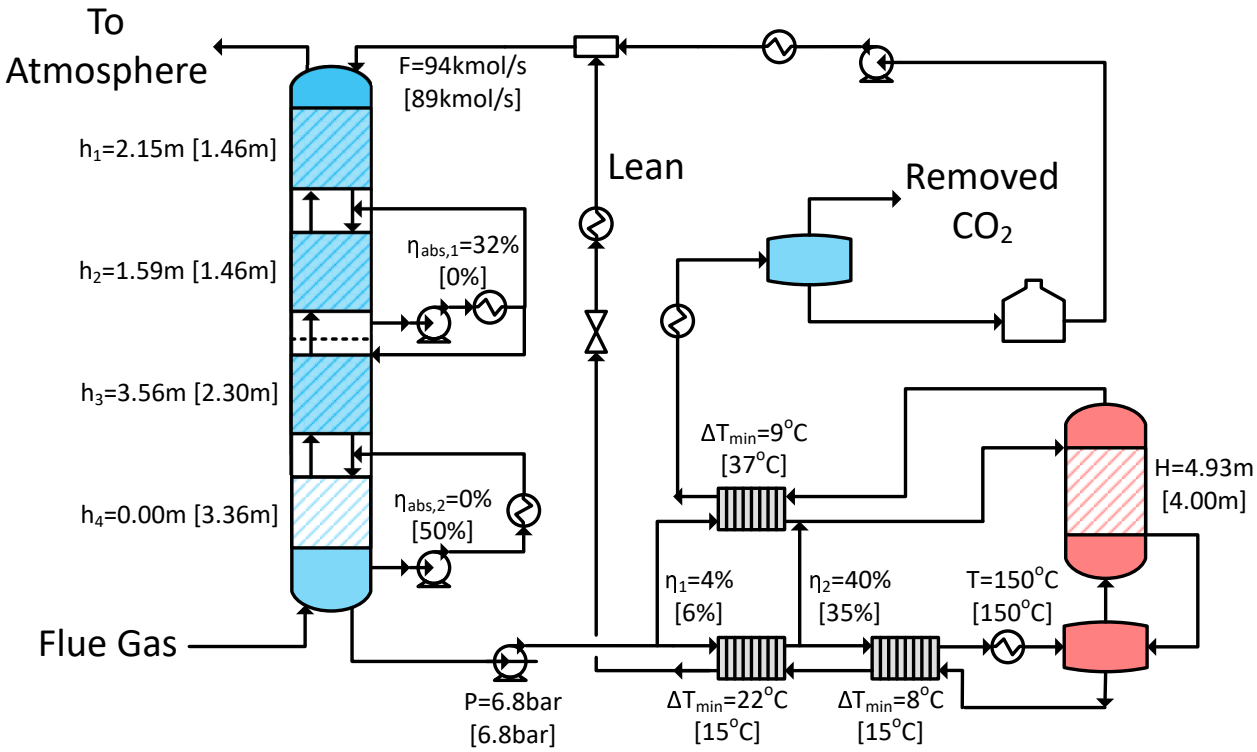

Figure 6.9: Base case and optimal design of the amine scrubbing process flowsheet. Base case values are shown in square brackets.

Regarding heat integration, the manipulation of heat exchanger sizes and bypasses slightly decreases the temperature of the warm bypass from 83°C in the base case to 79°C at the optimal point. The rich absorbent exits HX2 (S6) at 142°C, compared to 127°C in the base case, and the heating costs make up a much smaller percentage of overall annual cost. The drop in warm bypass temperature is compensated for by increasing the temperature of the cold bypass stream (S3) exiting HX3 from 64°C in the base case to 82°C at the optimal point. This increase is partially accomplished by decreasing the cold bypass ratio. The majority of the savings come from a reduction in heating costs, which are the main contributor to OPEX in the base case process. The heat use is decreased from 3.72 GJ to 2.5 7GJ per ton of CO_2, a number consistent with the best pilot plant results [154].

The profiles of key variables along the absorber are shown in Figure 6.10. Both the absorber and stripper have smaller volumes compared to the base case and make up a smaller proportion of the overall annual cost. Compared to the base case design, the height of the absorber is decreased at the optimal point, while removing the same amount of CO_2. The optimized absorber benefits from a decreased temperature bulge in the upper sections: the maximum liquid temperature in the base case absorber is 61.3°C, compared to 53.5°C in the optimal design. The bottom section of 2X packing with pumparound is not present at the optimal point, while 32% of the liquid at the bottom of the packed section 2 (Figure 6.5) is pumped back to the top of the section, lowering the temperature in the top section.

At the optimal point, the flash temperature reaches its maximum allowed value of 150°C to in turn maximize the vapor flow through the stripper column. The associated Lagrange multiplier at the optimum (shadow price) is 0.64 \$Million/°C. The shadow price p_s is defined as in (5.41), with J now being the total annual cost. The shadow price reveals that the local estimated effect of increasing the upper bound of temperature in the flash tank by 1°C is a decrease in total cost of \$0.64 million per year (\$0.155 per ton of CO_2 removed),

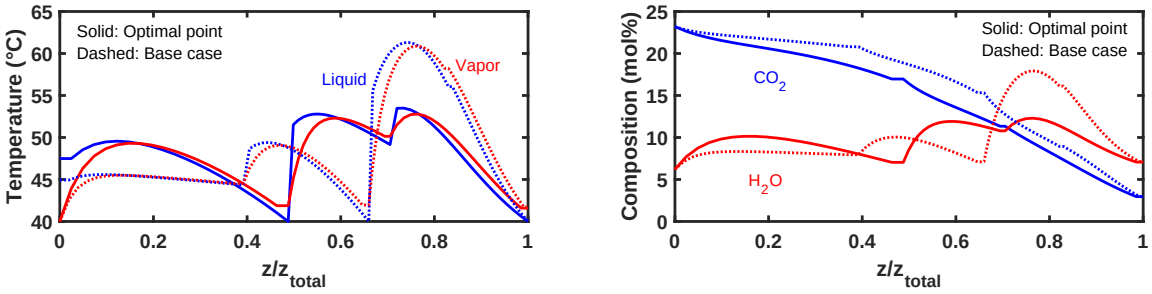

Figure 6.10: A comparison of absorber temperature and vapor composition profiles between base-case and optimal-point designs. The solid lines correspond to the optimal point design, while the dashed lines correspond to the base case design. The narrow "flat" regions between packed sections reflect the height correction described in Section 6.2.1.

providing insight into the benefits of improving the thermal stability of the solvent. This value can also guide the economic feasibility of operating the process at a higher temperature and constantly replacing part of the piperazine solvent that degrades. The percentage of CO_2 removed from the flue gas stream also reaches its lower bound, and the shadow price of CO_2 removal is 1.24 \$Million/%. Correspondingly, the local estimated effect of increasing the removal by 1% (0.046 million tons of CO_2 per year) is \$1.24 million per year, or \$27.13 per additional ton of CO_2 removed.

6.3.2 Case 2: Optimization for Increased CO_2 Removal

To examine the cost of increased removal for the same flue gas stream and hybrid-membrane process, the same optimization procedure was repeated with the CO_2 removal constrained to be at least 99%. Starting from the same initial guess, the optimal point was found in 50 NLP iterations, using 5287s of CPU time on the same desktop system. The optimal values of the decision variables and economic costs of the optimal design are shown as Case 2 in Tables 6.3–6.4. The decision variable values from Case 1 above are also reproduced for comparison. Compared to the optimal point found in Case 1, the heights of

Sections 1, 2, and 3 in the absorber are increased to achieve the requisite 99% CO_2 removal. Interestingly, the diameter of the absorber is slightly decreased in the solution, compared to the optimal point found for Case 1. The bottom section of MP2X packing with pumparound is again not present at the optimal point. The optimal design removes 99% of CO_2 from the flue gas stream, representing a capture of 4.53 million tons of CO_2 per year ($25.27 per ton of CO_2). The absorber and stripper both operate at a 70% flooding ratio, and the minumum approach temperatures of HX1, HX2, and HX3 are respectively 22.85°C, 6.76°C, and 9.28°C.

	Case 1	Case 2	Case 3
Intercooled Absorber			
Absorber Diameter (m)	13.59	13.35	12.94
Section 1 Height (m)	2.15	3.60	2.12
Section 2 Height (m)	1.59	2.38	1.52
Section 3 Height (m)	3.56	4.07	2.45
Section 4 Height (m)	0.00	0.00	0.00
Split 1 $\eta_{abs,1}$ (%)	32.0	20.3	47.7
Split 2 $\eta_{abs,2}$ (%)	0.0	0.0	0.0
Stripping Section			
Stripper Diameter (m)	8.74	9.56	7.29
Stripper Height (m)	4.92	4.75	4.49
Flash Temperature (°C)	150	150	150
Flash Pressure (bar)	5.80	5.68	5.52
Cold Bypass η_1 (%)	4.0	4.2	4.6
Warm Bypass η_2 (%)	40.0	44.9	47.6
Other Process Decisions			
Solvent (S17) Flow (kmol/s)	93.53	102.15	58.89
HX1 Area (m^2)	14157	14494	8971
HX2 Area (m^2)	37555	41342	21711
HX3 Area (m^2)	802	866	548

Table 6.3: Optimal-point values for the process design variables for Case 1 (base case optimization), Case 2 (increased removal), and Case 3 (lean flue gas).

Compared to the solution of Case 1, the minimum approach temperature of HX1 is further increased by 1.1°C, while that of HX2 is further decreased by 1.4°C. The minimum approach temperature of HX3 remains practically unchanged from Case 1 to Case 2. The manipulation of heat exchanger sizes and bypasses again slightly decreases the temperature of the warm bypass from 83°C in the base case to 80°C at the optimal point. The temperature

of the rich absorbent exiting HX2 (S6) is increased to 143°C, and the heating costs are again decreased compared to the base case. However, the heating costs are increased compared to the optimal solution of Case 1. The relative heat use at the optimal point increases slightly from 2.57 GJ per ton of CO_2 in Case 1 to 2.64 GJ in Case 2, indicating a slight tradeoff between process removal and efficiency in terms of heat use.

	Case 2		Case 3	
Capital Cost (PEC)	\$Million	contribution	\$Million	contribution
Absorber	17.91	15.6%	11.46	17.5%
Stripper	5.81	5.1%	4.05	6.2%
Exchangers	13.53	11.8%	7.45	11.4%
HX1	*3.46*		*2.14*	
HX2	*9.86*		*5.18*	
HX3	*0.21*		*0.13*	
Compressor	15.45	13.5%	8.08	12.4%
Steam Heater	1.45	1.3%	0.68	1.0%
Operating Cost	\$Million/yr	contribution	\$Million/yr	contribution
Heating	25.23	22.0%	14.63	22.4%
Cooling	8.29	7.2%	4.991	7.6%
Electricity	26.79	23.4%	14.00	21.4%
Totals	\$Million/yr	contribution	\$Million/yr	contribution
CAPEX	54.15	47.3%	31.71	48.5%
OPEX	60.31	52.7%	33.63	51.5%
TOTAL	**114.46**	**100.0%**	**65.34**	**100.0%**

Table 6.4: Process economics for Case 2 (increased removal) and Case 3 (lean flue gas).

The design for 99% removal requires a significantly larger absorber, and, consequently, the absorber comprises a larger percentage of the overall annualized cost. The solvent flow rate is also increased, resulting in larger process utility costs. The capital expenditure makes up a similar fraction of the overall cost as in the optimal design for 90% removal. As suggested by the increasing specific heat duty, the increase in CO_2 removal percentage results in a 3.3% increase in the average cost per ton of CO_2 captured. However, considering only the additional CO_2 captured (0.41 tons per year), the cost is \$33.27 per ton, a 36% increase in cost per ton of CO_2 compared to the base amount of CO_2 captured. The shadow price of increasing the upper bound of temperature increases to 0.78 \$Million/°C (\$0.172

per ton of CO_2 removed), suggesting that improving the thermal stability of the solvent has increasing benefits as CO_2 removal is increased. The percentage of CO_2 removed from the flue gas stream reaches its increased lower bound of 99%, and the shadow price of CO_2 removal is 2.96 $Million/%. The large increase in shadow price of CO_2 removal reveals that the cost of removing additional CO_2 also increases with CO_2 removal.

6.3.3 Case 3: Optimization for a Lean Flue Gas Stream

The same process was then optimized in a scenario reflecting CO_2 removal by an amine scrubbing process without membrane pre-treatment of the flue gas. A lean flue gas stream typical of post-combustion processes [109] was assumed, and the CO_2 removal was constrained to be at least 90%. The inlet flue gas flow rate was assumed to be 12.9 kmol/s, with molar composition of 13.2% CO_2, 9.5% H_2O, 73.3% Nitrogen, and 4% Oxygen. The optimal point was found in 50 NLP iterations, using 5091s of CPU time on the same desktop system, and is shown as Case 3 in Tables 6.3–6.4. The optimal design removes 90% of CO_2 from the flue gas stream, capturing 2.58 million tons of CO_2 per year ($25.33 per ton of CO_2). The absorber and stripper both operate at 70% flooding ratio, and the minimum approach temperatures in HX1, HX2, and HX3 are respectively 23.73°C, 5.95°C, and 10.39°C. The cost per ton of CO_2 captured is 3.5% higher that that of the hybrid membrane process.

The process design is significantly changed compared to the solution found in Case 1. The minimum approach temperature of HX1 is further increased by 1.9°C, and that of HX2 is further decreased by 2.2°C. The minimum approach temperature of HX3 is slightly higher compared to the solution of Case 1, but still much lower than the base case value. In contrast to the solutions for Cases 1 and 2, the temperature of the warm bypass is increased from the base case, up to 84°C at the optimal point. The warm and cold bypass streams have increased temperatures, as the solvent flow rate is significantly decreased. Heating costs

are still lowered with the increase in temperature of the rich absorbent exiting HX2 (S6) to 144°C. This high temperature is in part enabled by a large warm bypass of almost 48% at the optimal point. The heating costs comprise a similar proportion of the overall annual cost as in the solutions to Cases 1 and 2. The relative heat use at the optimal point increases dramatically from 2.57 GJ per ton of CO_2 in Case 1 to 2.96 GJ in Case 2, demonstrating the value of pre-treating the flue gas stream in terms of process efficiency.

Though it captures less CO_2, the design for the leaner (in terms of CO_2 concentration) flue gas requires a relatively large absorber, and the absorber comprises a much larger percentage of the overall annual cost. The bottom section of 2X packing is again not present at the optimal point, but the highest fraction of pumparound is required of all three cases to achieve the requisite CO_2 removal. The solvent flow rate is greatly decreased, and utility and HX costs are decreased. The shadow price of increasing the upper bound of temperature decreases to 0.43 \$Million/°C. While the magnitude of the shadow price decreases, the per unit shadow cost (\$0.167 per ton of CO_2 removed) remains similar to those found in the previous cases. Therefore the per unit value of improving the thermal stability of the solvent is relatively constant across all three optimal process designs. The percentage of CO_2 removed from the flue gas stream reaches its lower bound of 90%, and the shadow price of CO_2 removal is 0.83 \$Million/%. Correspondingly, the local estimated effect of increasing the removal by 1% (0.029 million tons of CO_2 per year) is \$0.83 million per year, or \$28.95 per additional ton of CO_2 removed. This suggests that the cost of removing additional CO_2 is larger for the case of a standalone amine scrubbing process compared to Case 1, further demonstrating the value of a higher concentration flue gas stream.

6.3.4 Discussion

All the optimal designs found in this chapter operate at several bounds: (i) the absorber and stripper always operate at the maximum 70% flooding ratio, which allows their diameters to be minimized; (ii) the temperature of the flash tank is always at its maximum allowed value of 150°C, a limitation imposed by the thermal stability of the amine-based solvent; and (iii) packed section 4 of the absorber, consisting of Mellapak 2X packing is never present. I note that the MP2X packing may introduce less pressure drop than Mellapak 250X, but pressure losses in the absorber were not considered in this work.

As shown in Table 6.3, the optimal designs found in this work are very different from one another. In a similar result, Cerrillo-Briones and Ricardez-Sandoval [50] found using a robust optimization approach that the optimal design of an absorber column can vary significantly when considering uncertainty in flue gas stream composition and/or desired CO_2 removal. At the optimal points, many decision variables do not reach an upper or lower bound (or they are unbounded), highlighting the necessity of computer-guided optimization procedures in designing the most economic process for a particular operating regime. The hybrid membrane process proposed by Freeman et al. [94] benefits from a higher inlet CO_2 concentration and significantly reduces the per unit cost of carbon capture. Consequently, the optimal design for the higher-concentration CO_2 flue gas costs 4% less (per ton of CO_2 captured) than the optimal design for a typical post-combustion flue gas stream. Changing the desired CO_2 removal percentage from the flue gas stream has a similar effect on the total cost per ton of CO_2. Increasing the fraction of CO_2 captured only increases the average per unit cost slightly, though the marginal cost of capturing additional CO_2 (added removal) is quite large.

6.4 Summary

This chapter introduced a pseudo-transient numerical approach for modeling rate-based packed columns, with particular application to packed absorbers and strippers. The models are generally applicable to rate-based column modeling and utilize pseudo-transient continuation. As a result of the latter, they are easily combined with other process unit models in the pseudo-transient process modeling framework in order to optimize process flowsheets incorporating rate-based column models. The proposed novel models lend themselves naturally to optimization applications, and I exploited this property to regress the models to predictions from a previously developed rigorous thermodynamic and kinetic model. I demonstrated the developments by considering a next-generation amine scrubbing process, finding large economic savings in comparison to a previously studied and heuristically optimized base case design.

6.5 Nomenclature

Table 6.5: Carbon capture process nomenclature.

Symbol	Description
α	loading (mol CO_2/mol alkalinity)
$\hat{\alpha}$	continuation parameter (dimensionless)
κ	thermal conductivity (MW/m K)
ρ	mass density (kg/m^3)
σ	surface tension (N/m)
τ	pseudo-transient time constant (pseudo-time)
A	area (m^2)
a_e	effective wetted area of packing (m^2/m^3)
a_p	specific area of packing (m^2/m^3)
C	concentration (mol/m^3)
C_p	specific heat capacity (MJ/mol K)
D	diffusion coefficient (cm^2/s)
F	molar flowrate (mol/s)
g	gravitational acceleration (m/s^2)
h	specific enthalpy (MJ/mol)
$\bar{h}$	convective heat transfer coefficient (MW/m^2 K)
K_g	overall mass transfer coefficient (mol/bar m^2 s)
k_g	gas mass transfer coefficient (mol/bar m^2 s)
k'_g	liquid mass transfer coefficient (mol/bar m^2 s)
L	height (m)
L_p	wetted perimeter (m)
N	material flux (mol/m^3 s)
N_H	enthalpy flux (MW/m^3)
P	pressure (bar)
P^*	equilibrium pressure (bar)
Q	heat duty (MJ)
R	gas constant (J/mol K)
t	pseudo-time
T	temperature (K)
U	overall heat transfer coefficient (MW/m^2 K)
v	superficial velocity (m/s)
x	liquid mole fraction (mol/mol)
y	vapor mole fraction (mol/mol)
z	height (m/m)

Table 6.6: Carbon capture process nomenclature (continued).

Superscripts	
C	cold-side
H	hot-side
L	liquid
PP	physical properties package
V	vapor
Subscripts	
$flood$	flooding
HX	heat exchanger
i	component
in	inlet
out	outlet

Chapter 7

Fast and Efficient Pseudo-Transient Flowsheet Simulation using Inertial Manifolds[†]

Chapters 4–6 showed that general, large-scale chemical process flowsheets involving complex unit operation models can be reliably simulated and optimized using pseudo-transient continuation (PTC) as a numerical method. Nevertheless, these chapters also revealed that the strategy remains computationally costly, with optimization CPU times in the order of hours. This can be attributed to the number of pseudo-time steps required to integrate the PTC model to steady state far exceeding the number of steps taken by a Newton-type solver (when a good initial point is available). Motivated by the above, in this chapter, I present a novel simulation approach for pseudo-transient process flowsheets, based on a hierarchical, multiply singularly perturbed formulation of the PTC dynamics. The proposed method allows a seamless transition from pseudo-time integration to efficient, Newton-type algebraic solvers, and it converges to the solution of the original algebraic system in a finite number of steps. Moreover, the complete *dynamic decomposition* of the PTC model eliminates the need to explicitly define the time constants $\boldsymbol{\tau}$. I demonstrate the computational advantages of the technique using two prototypical separation examples. The presentation in this chapter follows closely the study in Tsay and Baldea (2019) [254].

[†]The contents of this chapter are largely based on the following publication: C. Tsay and M. Baldea. Fast and efficient chemical process flowsheet simulation by pseudo-transient continuation on inertial manifolds. Comput. Method. Appl. M., 348:935–953, 2019. C.T. is the primary author of the manuscript.

7.1 Simple Example: Pressure-Enthalpy Flash

In this chapter, I illustrate fundamental concepts of pseudo-transient flowsheet modeling by considering a two-phase separator, a common process unit operation depicted in Figure 7.1.

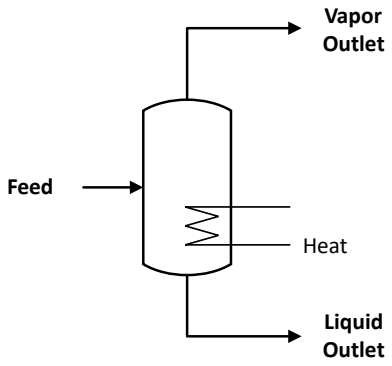

Figure 7.1: Diagram of a two-phase separator.

This device separates a multi-component inlet (feed) stream into two outlet streams, one liquid-phase and one gas-phase. To compute the flow rates and compositions of the two outlet streams, a pressure-enthalpy (PH) flash calculation of vapor-liquid phase equilibrium is performed. I now examine the mass balance, energy balance, and phase equilibrium relations that describe the PH flash at steady state. The model contains bilinear terms (the products of vapor or liquid flow rates, and the phase compositions) and a nonlinear equation

of state defining physical properties [190]:

$$0 = Mz_i - Vy_i - Lx_i, \ \forall \ i = 1, ..., n \tag{7.1}$$

$$0 = y_i - K_i x_i, \ \forall \ i = 1, ..., n \tag{7.2}$$

$$0 = \sum_{1}^{n} x_i - 1 \tag{7.3}$$

$$0 = \sum_{1}^{n} y_i - 1 \tag{7.4}$$

$$K_i = \frac{\phi_i^L}{\phi_i^V}, \ \forall \ i = 1, ..., n \tag{7.5}$$

$$0 = H_{in} - H_{out}^L - H_{out}^V + Q \tag{7.6}$$

where $i = 1, ..., n$ indexes the components present from lightest to heaviest, i.e., from the most to the least volatile. Each split fraction K_i is defined as the ratio between liquid and vapor fugacity coefficients, ϕ_i^L and respectively ϕ_i^V, which can be computed from an equation of state as a function of temperature T, pressure P, and composition $\boldsymbol{x}, \boldsymbol{y}$. The vapor and liquid flow rates and compositions (V, L, $\boldsymbol{x}$, $\boldsymbol{y}$) are initially unknown, while the inlet flow rate M, inlet compositions $\boldsymbol{z}$, pressure P, and heat added Q are known.

Although Section 4.2.2.1 presented a pseudo-transient method for reliably solving (7.1)–(7.6), the method relies on pressure-temperature (PT) flash calculations provided by a physical properties package. This example considers the case when PT flash calculations are not readily available. In this case, the model can be reformulated as a pseudo-transient

system with three state variables, each with an associated time constant [190]:

$$(\tau_1 y_1)\frac{dV}{d\hat{t}} = Mz_1 - Vy_1 - Lx_1 \tag{7.7}$$

$$0 = Mz_i - Vy_i - Lx_i,\ \forall\ i = 2, ..., n \tag{7.8}$$

$$0 = M - V - L \tag{7.9}$$

$$0 = y_i - K_i x_i,\ \forall\ i = 1, ..., n \tag{7.10}$$

$$0 = \sum_1^n x_i - 1 \tag{7.11}$$

$$0 = \sum_1^n y_i - 1 \tag{7.12}$$

$$\tau_2 \frac{dK_i}{d\hat{t}} = \frac{\phi_i^L}{\phi_i^V} - K_i,\ \forall\ i = 1, ..., n \tag{7.13}$$

$$(\tau_3 \frac{H_{in}}{T_0})\frac{dT}{d\hat{t}} = H_{in} - H_{out}^L - H_{out}^V + Q \tag{7.14}$$

This represents a prototypical pseudo-transient process model, as in (3.5)–(3.6), with $\mathbf{x}_d = [V, \boldsymbol{K}, T]$, $\mathbf{x}_s = [L, \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\phi}^L, \boldsymbol{\phi}^V, H_{out}^L, H_{out}^V]$, and $\boldsymbol{\tau} = [\tau_1, \tau_2, \tau_3]$. Note that the time constants in this chapter are numbered to reflect their position in the pseudo-transient dynamic hierarchy. The ODEs (7.7), (7.13), and (7.14) comprise $\mathbf{f}$, and the remaining equations (7.8)–(7.12) comprise $\mathbf{g}$. Equations (3.5)–(3.6) are reproduced below:

$$\mathbf{f}(\boldsymbol{\tau} \circ \dot{\mathbf{x}}_d, \mathbf{x}_d, \mathbf{x}_s, \mathbf{z}, \boldsymbol{\theta}) = 0 \tag{3.5 revisited}$$

$$\mathbf{g}(\mathbf{x}_d, \mathbf{x}_s, \mathbf{z}, \boldsymbol{\theta}) = 0 \tag{3.6 revisited}$$

By setting each time derivative term in (7.7)–(7.14) to zero, it can be easily verified that this pseudo-transient formulation satisfies the static equivalence property (Section 3.3.1). Additionally, once initial conditions for vapor flow, temperature, and vapor split fractions are provided, consistent initialization solely involves solving a full-rank linear system. Raoult's law (7.17) provides a reasonable initial guess for the vapor split fractions [190], and

the recommended initial conditions are:

$$V(\hat{t}=0) = \frac{M}{2} \tag{7.15}$$

$$T(\hat{t}=0) = \frac{T_{bub} + T_{dew}}{2} \tag{7.16}$$

$$K_i(\hat{t}=0) = \frac{P_i^{vap}(T)}{P} \tag{7.17}$$

where T_{bub} and T_{dew} are respectively the bubble and dew temperatures of the feed stream. Pattison and Baldea [190] showed that the model is stable—based on an eigenvalue analysis—when the dynamic mass balance is the fastest pseudo-transient dynamic ($\tau_1 << \tau_2, \tau_3$).

7.1.1 Simulation Results

The pseudo-transient model (7.7)–(7.14) was simulated with a feed stream composed of alkanes (ethane, propane, n-butane, n-pentane, and n-hexane). The feed temperature was assumed to be 250 K, the tank pressure to be 0.5 MPa, and the heat input to be 50 kW. The cubic Peng-Robinson equation of state was used for all physical properties, and the solution given by an initial simulation carried out in AspenPlus [9] is shown in Table 7.1. Note that the underlying model in AspenPlus may contain slight modifications compared with the given steady-state model (7.1)–(7.6). Simulations of the pseudo-transient model (7.7)–(7.14) in MATLAB R2015b [248]—as described below—produced the identical result, while simulation of the pseudo-transient model in gPROMS V4.1.0 [204] results in a slightly different steady-state solution. The minor differences between gPROMS and AspenPlus can be attributed to different implementations and parameter values for the equation of state: the steady-state temperatures found are respectively 287.5 K and 288.2 K.

The pseudo-transient model (7.7)–(7.14) was implemented in MATLAB and simulated using an explicit Runge-Kutta method. For comparison purposes, the same system was also implemented in the commercial package gPROMS [204], which uses the DASOLV package

Table 7.1: Stream flow rates and mole fractions for pressure-enthalpy flash.

Stream	Inlet (M)	Liquid (L)	Vapor (V)
Mole Flow (mol/s)	5	3.072	1.928
ethane (%)	30	11.55	59.39
propane (%)	20	16.67	25.30
n-butane (%)	20	25.93	10.55
n-pentane (%)	25	37.86	4.51
n-hexane (%)	5	7.98	0.25

to perform implicit time integration of DAE systems based on a variable time step/variable order Backward Difference Formula (BDF). The values used for simulation were $\tau_1 = 10^{-2}$, $\tau_2 = 10^{-1}$, and $\tau_3 = 10^2$, effectively creating a hierarchy of three time scales in the dynamics of the pseudo-transient model. The parameters given by Cash and Karp [47] were selected for explicit fourth-order Runge-Kutta (RK4) time integration.

Although only the steady-state solution is important for a pseudo-transient model, explicit time integration methods may result in unstable trajectories for the state variables if the size of each time integration step is not carefully selected. The system was simulated with an adaptive step size calculated using an evolutionary PID (proportional-integral-derivative) controller. The estimated error was computed using the difference between fifth-order (RK5) and embedded fourth-order (RK4) Runge-Kutta integration. If the estimated error is greater than a prescribed tolerance, the step size is reduced:

$$\Delta\hat{t}_n = \left|\frac{e_{tol}}{e_n}\right|^{0.2} \Delta\hat{t}_{n,0} \tag{7.18}$$

$$\Delta\hat{t}_{min} \leq \Delta\hat{t}_{n+1} \leq \Delta\hat{t}_{max} \tag{7.19}$$

where e_n denotes the estimated error in $\mathbf{x}_d$, e_{tol} denotes the prescribed tolerance, $\Delta\hat{t}_{n,0}$ is the initial guess of the step size, and $\Delta\hat{t}_n$ is the updated step size. The exponent in (7.18) was set to be 0.2 since the error for an RK4 method is $O(\Delta\hat{t}^5)$.

The initial guess of the step size for the ensuing integration step is then smoothly adjusted using an evolutionary PID controller [144]:

$$\Delta\hat{t}_{n+1} = \left(\frac{e_{n-1}}{e_n}\right)^{k_P}\left(\frac{e_{tol}}{e_n}\right)^{k_I}\left(\frac{e_{n-1}^2}{e_n e_{n-2}}\right)^{k_D}\Delta\hat{t}_n \tag{7.20}$$

The empirical PID tuning ($k_P = 0.075$, $k_I = 0.175$, $k_D = 0.01$) given by Kuzmin [144] was used for all simulations in this chapter. The computational efficiency of pseudo-time integration can be analyzed by examining the rate at which the residual (the sum of the 2-norms of all pseudo-time derivatives) approaches zero. Figure 7.2 shows the residual and the time integration step length for the tested integration methods: the implicit DASOLV package and the described explicit RK4 integration procedure. The interested reader is referred to [104] for a thorough discussion regarding PID control and other methods for integration step size control in explicit Runge-Kutta methods.

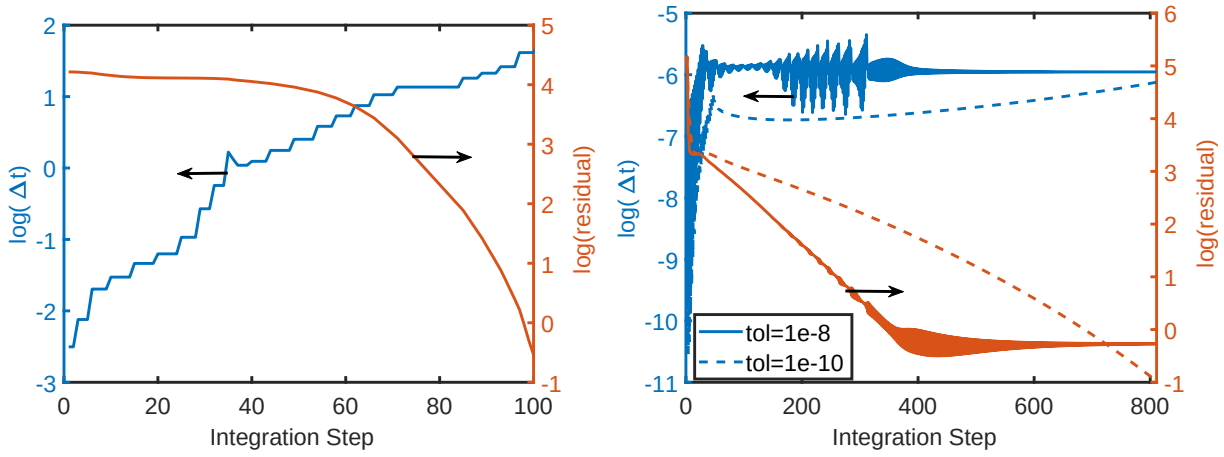

Figure 7.2: Time step and residual during pseudo-transient pressure-enthalpy flash simulation. Results from DASOLV are shown on the left and from RK4 are shown on the right.

For both methods, the integration step sizes increase as the residual decreases, and the rate of convergence accelerates. Recall that an infinitely long time step is mathematically equivalent to applying an undamped Newton step on the original algebraic system [128] (see

also Chapter 3). Instead, taking smaller "time steps" allows for a smooth approach to the steady-state solution. Integration with the implicit DASOLV algorithm allowed much larger time steps to be taken during integration and consequently required fewer integration steps to reach steady state [111]. However, each time step of implicit integration requires more computational effort, and the computation times for the two methods are of the same order of magnitude. Simulation with DASOLV (using the default solver settings) in gPROMS required 2s on a Windows 7 laptop computer running at 3.00 GHz, while solution with the RK4 method required 6.7s and 7.8s, for integration tolerances of 10^{-8} and 10^{-10} respectively.

To analyze the convergence advantages of the PT reformulation over the original algebraic model, the solution of both models was attempted with various initial guesses for temperature T and vapor split fraction K_i. A set of initial guesses was created by introducing an error between -20% and +20% with respect to the true values of T and K_i at steady state (the same error was used for all K_i values in Figure 7.3). For the algebraic model, the true steady-state values of all other variables were used as initial guesses, an advantage that is not ordinarily available. For the pseudo-transient model, other initial guesses are not required. Solution of the algebraic model was attempted using the `fsolve` function in MATLAB, which solves nonlinear systems using the Levenberg-Marquadt algorithm. This iterative algorithm [127] interpolates step directions between gradient descent and Newton steps, allowing convergence from a wider range of initial guesses than the standard Newton method. As expected, the pseudo-transient model is robust and converges from a wide range of initial guesses. However, when the algebraic model converged, it required fewer iterations. Since simulation from most initial guesses converged in only a few iterations (Figure 7.3, left), initial guesses that required the maximum number of iterations (set to 500) are not likely to converge. The convergence basin for even this simple algebraic model is difficult to predict due to nonlinearity, and furthermore, Figure 7.3 (left) shows that the basin of

convergence might not be a connected set. Although the pseudo-transient model requires more integration steps than the algebraic model requires Levenberg-Marquadt iterations, in most cases the system eventually converges to the correct steady state solution.

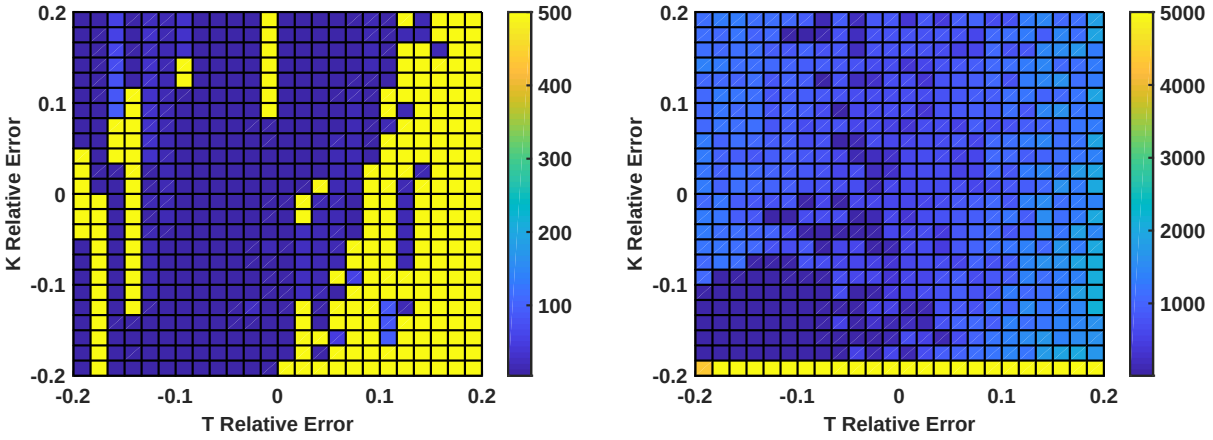

Figure 7.3: Convergence basin of pressure-enthalpy flash simulation using algebraic model (left) and using pseudo-transient model (right). The algebraic results depict the number of Levenberg-Marquadt iterations taken, and the pseudo-transient results depict the number of pseudo-time integration steps taken.

7.2 Quasi-Steady-States in Pseudo-Transient Models

The potential fast convergence of algebraic solvers suggests that a natural strategy for process flowsheet simulation would be to use PTC to evolve the flowsheet variables until they reach the convergence basin for an algebraic solver, then switch to the algebraic solver using the values of the states and algebraic variables at the end of pseudo-time integration as initial guesses. This concept is difficult to implement in practice, as the convergence basin for a general non-linear algebraic system is impossible to discern *a priori*, and the point where the PT model should be "switched" to the original algebraic model is therefore unknown. Ma et al. [158] found that the computational time required for PT process optimization could be reduced by 70-90% by simply attempting algebraic flowsheet simulation at each

optimization iteration. If the algebraic solver does not converge, the standard PTC strategy is used. Although this strategy allows switching to an algebraic solver, it may involve many unnecessary Newton iterations on the full model (3.1) without convergence. These added iterations are costly, as a Newton step requires computing the Jacobian of the system—$O(n^2)$ operations—whereas an explicit integration step only requires $O(n)$ operations.

I now present a singular perturbation analysis of pseudo-transient systems and propose a systematic approach for transitioning between the PTC model and the original algebraic equations. The scheme employs a quasi-steady-state assumption for the dynamics in certain time scales. I use the above strategy (periodically checking the algebraic convergence of the system), but the algebraic system size is greatly reduced by only "switching" a subset of the equations from pseudo-transient to algebraic. The fundamental concept of pseudo-transient modeling is to dynamically decouple the algebraic equations underlying a process flowsheet for reliable computation. Although PTC widens the set of initial guesses that can be used, convergence is still dependent on the selection of $\boldsymbol{\tau}$; this is the primary challenge in pseudo-transient process modeling. Poorly chosen time constants can lead to unstable systems that fail to converge, or ill-conditioned systems that cannot be easily simulated.

7.2.1 Dynamically Decoupled Models

When the time constants are selected properly, the model equations are dynamically decoupled. As presented in the Simple Example, the values of the three time constants differ by several orders of magnitude, and the dynamic mass balance, phase equilibrium equations, and dynamic heat balance evolve in different time scales. In general, a pseudo-transient model (3.5)–(3.6) with k individual time constants represents a multiple singularly perturbed system in standard form, with dynamics partitioned into a hierarchy of time scales.

Assuming explicit first-order dynamics, the system can be described as:

$$\tau_1 \dot{\mathbf{x}}_{d,1} = \mathbf{f}_1(\mathbf{x}_d, \mathbf{x}_s, \boldsymbol{\theta}),\ \dot{\mathbf{x}}_{d,1} \sim \mathcal{O}(\tau_1^{-1}\mathbf{x}_{d,1}) \qquad fastest \tag{7.21}$$

$$\tau_2 \dot{\mathbf{x}}_{d,2} = \mathbf{f}_2(\mathbf{x}_d, \mathbf{x}_s, \boldsymbol{\theta}),\ \dot{\mathbf{x}}_{d,2} \sim \mathcal{O}(\tau_2^{-1}\mathbf{x}_{d,2}) \qquad \downarrow \tag{7.22}$$

$$\vdots \tag{7.23}$$

$$\tau_k \dot{\mathbf{x}}_{d,k} = \mathbf{f}_k(\mathbf{x}_d, \mathbf{x}_s, \boldsymbol{\theta}),\ \dot{\mathbf{x}}_{d,k} \sim \mathcal{O}(\tau_k^{-1}\mathbf{x}_{d,k}) \qquad slowest \tag{7.24}$$

$$\mathbf{g}(\mathbf{x_d}, \mathbf{x_s}, \boldsymbol{\theta}) = 0 \qquad steady\ state\ (algebraic) \tag{7.25}$$

with $\tau_1^{-1} >> \tau_2^{-1} >> ... >> \tau_k^{-1}$ to dynamically decouple the first-order dynamic equations $\mathbf{f}_1, \mathbf{f}_2, ..., \mathbf{f}_k$. The steady state of the first-order differential equation(s) $\mathbf{f}_i$ corresponds to some algebraic constraint(s) of the original model $\mathbf{f}_{ss,i}$, such that $\mathbf{f}_{ss,i} = 0$ when $\mathbf{f}_i = 0$. Defining a hierarchy of time scales $\theta_i = 1/\tau_i$, the equations can be rewritten as:

$$\frac{d\mathbf{x}_{d,1}}{d\theta_1} = \mathbf{f}_1(\mathbf{x}_d, \mathbf{x}_s, \boldsymbol{\theta}) \sim \mathcal{O}(\mathbf{x}_{d,1}) \tag{7.26}$$

$$\frac{d\mathbf{x}_{d,2}}{d\theta_2} = \mathbf{f}_2(\mathbf{x}_d, \mathbf{x}_s, \boldsymbol{\theta}) \sim \mathcal{O}(\mathbf{x}_{d,2}) \tag{7.27}$$

$$\vdots \tag{7.28}$$

$$\frac{d\mathbf{x}_{d,k}}{d\theta_k} = \mathbf{f}_k(\mathbf{x}_d, \mathbf{x}_s, \boldsymbol{\theta}) \sim \mathcal{O}(\mathbf{x}_{d,k}) \tag{7.29}$$

$$\mathbf{g}(\mathbf{x}_d, \mathbf{x}_s, \boldsymbol{\theta}) = 0 \tag{7.30}$$

Since $\tau_i^{-1} >> \tau_{i+1}^{-1}$, and equivalently, $\theta_i >> \theta_{i+1}$, the system dynamics can be analyzed using singular perturbation arguments by defining the ratio between time scales $\frac{\theta_{i+1}}{\theta_i} = \frac{\tau_i}{\tau_{i+1}}$ as the perturbation variables ϵ_i. Assuming these ratios are all equal, or $\epsilon_1, ..., \epsilon_{k-1} = \epsilon$, with

$0 < \epsilon << 1$, the system dynamics in time scale θ_{i+1} are:

$$\epsilon^i \frac{d\mathbf{x}_{d,1}}{d\theta_{i+1}} = \mathbf{f}_1(\mathbf{x}_d, \mathbf{x}_s, \boldsymbol{\theta}) \to 0 \tag{7.31}$$

$$\vdots \tag{7.32}$$

$$\epsilon \frac{d\mathbf{x}_{d,i}}{d\theta_{i+1}} = \mathbf{f}_i(\mathbf{x}_d, \mathbf{x}_s, \boldsymbol{\theta}) \to 0 \tag{7.33}$$

$$\frac{d\mathbf{x}_{d,i+1}}{d\theta_{i+1}} = \epsilon^0 \mathbf{f}_{i+1}(\mathbf{x}_d, \mathbf{x}_s, \boldsymbol{\theta}) \to 0 \tag{7.34}$$

$$\vdots \tag{7.35}$$

$$\frac{d\mathbf{x}_{d,k}}{d\theta_{i+1}} = \epsilon^{k-i-1} \mathbf{f}_k(\mathbf{x}_d, \mathbf{x}_s, \boldsymbol{\theta}) \to 0 \tag{7.36}$$

$$\mathbf{g}(\mathbf{x}_d, \mathbf{x}_s, \boldsymbol{\theta}) = 0 \tag{7.37}$$

Considering the limit $\epsilon \to 0$, the time scales faster than θ_{i+1} approach zero, and $\epsilon^{i-j+1} \frac{d\mathbf{x}_{d,j}}{d\theta_{i+1}} \to 0 \ \forall \ j = 1, ..., i$. The differential equations in (7.31)–(7.33) are thus approximately at a quasi-steady state, and $\mathbf{f}_1, .., \mathbf{f}_i \approx 0$, or, equivalently, $\mathbf{f}_{ss,1}, .., \mathbf{f}_{ss,i} \approx 0$. Thus, the pseudo-transient dynamics are naturally partitioned into a fast component at pseudo-steady state and a set of slower dynamics, which continue to evolve on slower time scales. In the same θ_{i+1} time scale, the system dynamics can be approximated:

$$\frac{d\mathbf{x}_{d,i+1}}{d\theta_{i+1}} = \epsilon^0 \mathbf{f}_{i+1}(\mathbf{x}_d, \mathbf{x}_s, \boldsymbol{\theta}) \tag{7.38}$$

$$\vdots \tag{7.39}$$

$$\frac{d\mathbf{x}_{d,k}}{d\theta_{i+1}} = \epsilon^{k-i-1} \mathbf{f}_k(\mathbf{x}_d, \mathbf{x}_s, \boldsymbol{\theta}) \approx 0 \tag{7.40}$$

$$\mathbf{f}^i_{QSS}(\mathbf{x}_d, \mathbf{x}_s, \boldsymbol{\theta}) = \{\mathbf{f}_{ss,1}, \mathbf{f}_{ss,2}, ..., \mathbf{f}_{ss,i}, \mathbf{g}\} = 0 \tag{7.41}$$

The quasi-steady-state equations $\mathbf{f}^i_{QSS} = 0$ specify an *inertial manifold* on which the slower dynamics $\mathbf{f}_{i+1}, ..., \mathbf{f}_k$ continue to evolve.

In the Simple Example (7.7)–(7.14), there are three separate (pseudo-transient) time scales ($k = 3$). τ_1 governs the rate at which the *mass* balance (7.7) variables evolve, which

corresponds to the fastest dynamics in the proposed approach. τ_2 governs the rate of change for the phase *equilibrium* equations (7.13)) are solved, corresponding to dynamics of intermediate speed. Lastly, τ_3 governs the rate at which the *heat* (energy) balance (7.14)) variables evolve, corresponding to the slowest dynamics. For the simulations shown in Figure 7.2, the aggregate residual can be partitioned into sub-residuals for the mass, equilibrium, and heat balance equations, respectively, as the absolute values of $\mathbf{f}_1$, $\mathbf{f}_2$, and $\mathbf{f}_3$. Figure 7.4 shows the contribution of each of these terms (each scaled by its associated time constant) to the total residual (Figure 7.2).

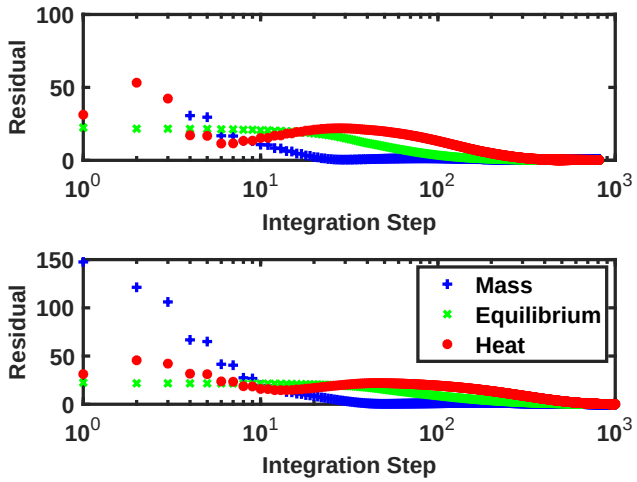

Figure 7.4: Behavior of mass, phase equilibrium, and heat pseudo-transient equations during time integration. The top plot shows simulation with an RK4 integration tolerance of 10^{-8}, and the bottom plot shows simulation with a tolerance of 10^{-10}.

It can be seen that—with proper selection of time constants for dynamic decoupling—the pseudo-transient equations reach quasi-steady state in the order of their dynamic hierarchy. In this example, the dynamic mass balance reaches a quasi-steady state, and the phase equilibrium dynamic equation is solved on the inertial manifold approximately satisfying the steady-state mass balance. Similarly, once the phase equilibrium dynamic equations reach

quasi-steady state, the dynamic heat balance is solved on the inertial manifold satisfying the phase equilibrium and mass balance equations. Once a particular pseudo-transient equation reaches its inertial manifold, the system remains approximately on this equilibrium manifold while the slower pseudo-transient dynamics evolve.

7.2.2 Simulation on Inertial Manifolds

An alternative visualization of the system evolution on inertial manifolds is shown in Figure 7.5. The plots show the evolution of dynamics in adjacent time scales in terms of their residuals, starting from several initial conditions located in the same region of the phase plane with respect to the true steady-state solution. The initial conditions were set by adding 5%, 10%, and 15% error to the steady-state values of all dynamic variables, $\mathbf{x}_d$.

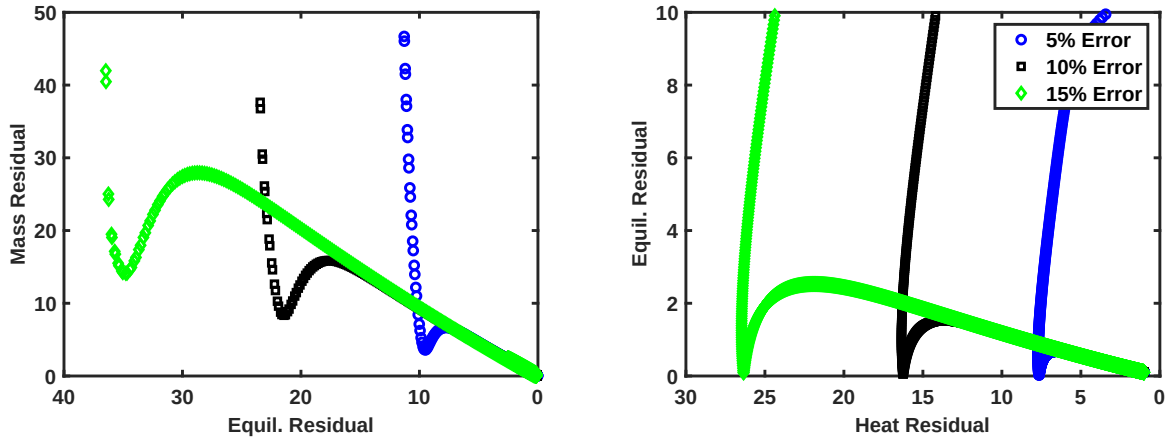

Figure 7.5: Residuals of the mass, phase equilibrium, and heat pseudo-transient equations during time integration (RK4 tolerance of 10^{-8}). The left plot shows the mass and phase equilibrium residuals. The right plot shows the phase equilibrium and heat residuals.

It can be seen that the dynamic mass balance residual $\mathbf{f}_1$ evolving in the fast time scale θ_1 initially decreases fast while the phase equilibrium residual (evolving in θ_2) is relatively unchanged. At a point in the simulation, the two residuals begin to decrease at a similar

rate, suggesting that the system reaches the inertial manifold defined by the quasi-steady state of the dynamic mass balance. A similar relationship can be seen between the dynamic phase equilibrium equations and the dynamic heat balance in the slowest time scale θ_3, confirming that the inertial manifolds are reached sequentially. Each point plotted in Figure 7.5 represents an integration step, and it can be seen that fewer integration steps are needed to solve the dynamic mass balance compared to the equilibrium equations or heat balance. This is caused by system stiffness: the dynamic mass balance always evolves in the fast time scale θ_1, and appropriately small integration step sizes must be selected even after the dynamic mass balance has reached a quasi-steady state.

Novel Algorithm. I propose to enforce the algebraic quasi-steady-state assumption for the equations in each time scale once they reach their respective inertial manifolds by exploiting the known hierarchy of time scales in PT flowsheets. Considering the above general pseudo-transient system with k hierarchical time scales, the dynamic equations should reach steady state sequentially, in order from fastest to slowest ($\mathbf{f}_1$ to $\mathbf{f}_k$). Introducing algebraic quasi-steady-state equations exploits the hierarchy of inertial manifolds present in multiply singularly perturbed dynamic systems. Moreover, the PT differential equations need only be assigned to a hierarchy of time scales, and *there is no need to explicitly define the values of the time constants*, removing most of the remaining empirical elements from the PT simulation approach. Each (set of) dynamic equation(s) $\dot{\mathbf{x}}_{d,i} = \mathbf{f}_i$ is integrated to quasi-steady state in its original unscaled form, reaching the inertial manifold defined by $\mathbf{f}_{ss,i} = 0$. The process can be broken down into k steps, involving two hyper-parameters, nPT and $nAlg$,:

Step 0: *(Initialization)* Solve the system of equations $\mathbf{f}^0_{QSS} = \{\mathbf{g}\}$ for $\mathbf{x}_s$ with the values of $\mathbf{x}_d$ fixed to their initial conditions. Note this step is equivalent to the consistent initialization step of pseudo-transient DAE systems. Based on the formulation of pseudo-transient process models, this step typically represents solving a full-rank set of linear equations.

Step 1: Solve the system of equations $\mathbf{f}^1_{QSS} = \{\mathbf{f}_{ss,1}, \mathbf{g}\}$ for $\{\mathbf{x}_{d,1}, \mathbf{x}_{ss}\}$ using an algebraic solver, with the values of $\mathbf{x}_d \backslash \{\mathbf{x}_{d,1}\}$ fixed to their initial conditions. If the solver does not converge in $nAlg$ iterations, update the initial guess of $\mathbf{x}_{d,1}$ for algebraic solution by integrating the dynamic equation $\dot{\mathbf{x}}_{d,1} = \mathbf{f}_1(\mathbf{x}_d, \mathbf{x}_s, \theta)$ forward nPT steps. Repeat ***Step 1*** until $\mathbf{f}^1_{QSS}$ can be solved within a specified tolerance for $\{\mathbf{x}_{d,1}, \mathbf{x}_s\}$.

Step 2: Solve the system of equations $\mathbf{f}^2_{QSS} = \{\mathbf{f}_{ss,1}, \mathbf{f}_{ss,2}, \mathbf{g}\}$ for $\{\mathbf{x}_{d,1}, \mathbf{x}_{d,2}, \mathbf{x}_{ss}\}$ using an algebraic solver, with the values of $\mathbf{x}_d \setminus \{\mathbf{x}_{d,1}, \mathbf{x}_{d,2}\}$ fixed to their initial conditions. If the solver does not converge in $nAlg$ iterations, update the initial guess of $\mathbf{x}_{d,2}$ for algebraic solution by integrating the dynamic equation $\dot{\mathbf{x}}_{d,2} = \mathbf{f}_2(\mathbf{x_d}, \mathbf{x_s}, \theta)$ forward nPT steps. At each step, use ***Step 1*** to update $\{\mathbf{x}_{d,1}, \mathbf{x}_s\}$ for the updated initial guess of $\mathbf{x}_{d,2}$. Repeat ***Step 2*** until $\mathbf{f}^2_{QSS}$ can be solved within a specified tolerance for $\{\mathbf{x}_{d,1}, \mathbf{x}_{d,2}, \mathbf{x}_s\}$.

$\vdots$

Step k-1: Solve the system of equations $\mathbf{f}^{k-1}_{QSS} = \{\mathbf{f}_{ss,1}, \mathbf{f}_{ss,2}, ..., \mathbf{f}_{ss,k-1}, \mathbf{g}\}$ for $\{\mathbf{x}_{d,1}, \mathbf{x}_{d,2}, ..., \mathbf{x}_{k-1}, \mathbf{x}_{ss}\}$ using an algebraic solver, with the values of $\mathbf{x}_d \setminus \{\mathbf{x}_{d,1}, \mathbf{x}_{d,2}, \mathbf{x}_{d,k-1}\}$ fixed to their initial conditions. If the solver does not converge in $nAlg$ iterations, update the initial guess of $\mathbf{x}_{d,k-1}$ for algebraic solution by integrating the dynamic equation $\dot{\mathbf{x}}_{d,k-1} = \mathbf{f}_{k-1}(\mathbf{x}_d, \mathbf{x}_s, \theta)$ forward nPT steps. At each step, use ***Step k-2*** to update $\{\mathbf{x}_{d,1}, \mathbf{x}_{d,2}, ..., \mathbf{x}_{d,k-2}, \mathbf{x}_s\}$ for the updated initial guess of $\mathbf{x}_{d,k-1}$. Repeat ***Step k-1*** until $\mathbf{f}^{k-1}_{QSS}$ can be solved within a specified tolerance for $\{\mathbf{x}_{d,1}, \mathbf{x}_{d,2}, ..., \mathbf{x}_{d,k-1}, \mathbf{x}_s\}$.

Step k: Solve the system of equations $\mathbf{f}_{QSS}^{k} = \{\mathbf{f}_{ss,1}, \mathbf{f}_{ss,2}, ..., \mathbf{f}_{ss,k}, \mathbf{g}\}$ for $\{\mathbf{x}_d, \mathbf{x}_{ss}\}$ using an algebraic solver. If the solver does not converge in $nAlg$ iterations, update the initial guess of $\mathbf{x}_{d,k}$ for algebraic solution by integrating the dynamic equation $\dot{\mathbf{x}}_{d,k} = \mathbf{f}_k(\mathbf{x}_d, \mathbf{x}_s, \theta)$ forward nPT steps. At each step, use ***Step k-1*** to update $\{\mathbf{x}_{d,1}, ..., \mathbf{x}_{d,k-1}, \mathbf{x}_s\}$ for the updated initial guess of $\mathbf{x}_{d,k}$. Repeat ***Step k*** until $\mathbf{f}_{QSS}^{k}$ can be solved within a specified tolerance for $\{\mathbf{x}_d, \mathbf{x}_s\}$.

Proposition 1. If a pseudo-transient flowsheet model is solvable (i.e., $\frac{\partial \mathbf{f}}{\partial \mathbf{x}}$ is full rank) then the partitions $\mathbf{f}_{QSS}^{1}, ..., \mathbf{f}_{QSS}^{k}$ of the same model are solvable, regardless of the partitioning selected for ***Step 0*** through ***Step k***.

Proof. Pseudo-transient flowsheet models are index-1 DAE systems, and consistent initialization requires that the algebraic system $\mathbf{g}$ be solvable given values for $\mathbf{x}_d$ (the Jacobian of $\mathbf{g}$ w.r.t. $\mathbf{f_s}$ is full rank). The derivative of each pseudo-transient flowsheet equation f_d w.r.t. its associated state variable x_d is non-zero, such that f_d eventually reaches steady state for some value of x_d. This ensures the Jacobian of $\mathbf{f}_{ss,i}$ w.r.t. $\mathbf{x}_{d,i}$ has non-zero diagonal entries. It can therefore be verified that the partition $\mathbf{f}_{QSS}^{i}$ is solvable if all equations in $\{\mathbf{f}_{ss,1}, ..., \mathbf{f}_{ss,i}\}$ are linearly independent. This property is satisfied if the original system $\mathbf{f}_{ss}$ is solvable (full-rank). □

The final system of equations $\mathbf{f}_{QSS}^{k} = \{\mathbf{f}_{ss,1}, \mathbf{f}_{ss,2}, ..., \mathbf{f}_{ss,k-1}, \mathbf{g}\}$ is equivalent to the original algebraic steady-state model $\mathbf{f}_{ss}$ (3.1), and the flowsheet simulation is complete. This procedure converts the pseudo-transient dynamic equations $\mathbf{f}_i,\ i = 1, ..., k$ back to the corresponding algebraic equations $\mathbf{f}_{ss,i},\ i = 1, ..., k$ in a sequential order when it is mathematically advantageous to do so (i.e., when the corresponding algebraic equations can be solved). The steady-state equations are reverted to pseudo-transient form and subjected to PTC when the algebraic solver fails to converge. This gradual conversion has two major advantages. First, numerical integration of the slower dynamics alone can be carried out

with larger step sizes, as simultaneous integration of systems whose dynamics evolve over multiple time scales (stiff systems) is costly [111]. Second, the need to empirically define time constants $\boldsymbol{\tau}$ for the pseudo-transient system is eliminated. As is standard for pseudo-transient models, the underlying equations must be partitioned into a dynamic hierarchy. However, the dynamics in each time scale of the hierarchy are inherently decoupled from the others, and the user does not need to explicitly select time constants.

Proposition 2. If a pseudo-transient flowsheet model is asymptotically stable, then simulation with the algorithm described in ***Steps 0–k*** will converge to the steady-state solution in a finite number of steps for any selection of nPT and $nAlg$ greater than or equal to 1.

Proof. Pseudo-transient models are defined such that each set of dynamic equations $\frac{d\mathbf{x}_{d,i}}{d\theta_i} = \mathbf{f_i}$ is asymptotically stable around its associated steady-state solution $\mathbf{f}_{ss,i} = 0$ [190]. For some process models, this stability requires that the faster pseudo-transient equations be (approximately) at steady state, $\mathbf{f}_{ss,j} = 0 \ \forall \ j = 1, ..., i-1$ [190, 261], which is trivially satisfied by the proposed algorithm. Following the proposed algorithm, dynamic stability in the fastest pseudo-transient time scale θ_1 is contingent only on the dynamics of $\dot{\mathbf{x}}_{d,1} = \mathbf{f}_1$ itself. Since $\dot{\mathbf{x}}_{d,1} = \mathbf{f}_1$ is asymptotically stable, $\dot{\mathbf{x}}_{d,1}$ will reach steady state in a finite number of pseudo-transient integration steps nPT. A proportional number of algebraic steps $nAlg$ will be taken, as solution with $nAlg$ algebraic steps is attempted after every nPT integration steps. Note that $\dot{\mathbf{x}}_{d,1}$ may only be stable for certain choices of initial condition, as reflected by the PTC convergence basin (Figure 7.3, right).

Dynamic stability of the differential equation(s) in a subsequent slower time scale $i \geq 2$ is contingent on the dynamics of $\mathbf{f}_i$, and also on convergence of $\mathbf{f}_{ss,j} = 0 \ \forall \ j = 1, ..., i-1$. Here, at each integration step, $\mathbf{x}_{d,i}$ is updated from its current value $\mathbf{x}_{d,i}$ to $\mathbf{x}^+_{d,i}$, with the update $\boldsymbol{\eta} = \mathbf{x}^+_{d,i} - \mathbf{x}_{d,i}$ based on the dynamics $\dot{\mathbf{x}}_{d,i} = \mathbf{f}_i$. Each algebraic equation $\mathbf{f}_{ss,j}$ is then

reverted to pseudo-transient form $\mathbf{f}_j$ and integrated to steady state if needed for convergence. Thus, the convergence of $\mathbf{f}_{ss,j} = 0$ in a finite number of steps depends only on the dynamic stability of $\mathbf{f_j}$ to its steady state $\mathbf{f}_{ss,j}(\mathbf{x}_{d,i} + \boldsymbol{\eta}, ...) = 0$ from the initial condition given by $\mathbf{f}_{ss,j}(\mathbf{x}_{d,i}, ...) = 0$. The inertial manifold corresponding to each equation $\mathbf{f}_{ss,j} = 0$ must therefore be a local attractor to small perturbations $\boldsymbol{\eta}$.

This property is required for stability of the dynamically decoupled (singularly perturbed) pseudo-transient system, as described in the assumptions for *Theorem 2.1* in [137]. Intuitively, the faster dynamics $\mathbf{f}_j \; \forall \; j = 1, ..., i-1$ must approximately reach steady state when $\mathbf{x}_{d,i}$ is integrated forward to $\mathbf{x}^+_{d,i} = \mathbf{x}_{d,i} + \boldsymbol{\eta}$. Given that each set of dynamics $\mathbf{f}_i$ is stable at its steady-state solution, time integration on each hierarchical time scale will reach quasi-steady state in a finite number of PT integration steps. □

7.2.3 Relevance to Pseudo-Transient Flowsheet Optimization

The proposed sequential quasi-steady-state algorithm can be used to expedite the solution of a library of pseudo-transient models [190, 195, 260] that rely on the dynamic decoupling concept. Pseudo-transient models can also be used for process flowsheet optimization via a time-relaxation-based steady-state optimization procedure [268]. Here, the pseudo-transient process flowsheet is initialized using the initial guesses for decision and state variables as initial conditions, and the model is integrated until it reaches steady state. Optimization-relevant gradients (the gradients of objective function and inequality constraints with respect to the decision variables) computed using symbolic or automatic differentiation are then supplied to an optimization solver that updates the decision variables for the next iteration, and the process is repeated until some pre-determined optimality criteria are met. For all iterations after the first, the process flowsheet is initialized using the steady state solution of the previous iteration as the initial condition.

Since the pseudo-transient time integration is used solely as a mathematical device to simulate the process flowsheet at each optimization iteration, it can easily by substituted with the proposed algorithm. In Chapters 4–6, the bulk of the computation time required for process optimization was spent on initializing the process flowsheet by integrating the pseudo-transient model for each iteration, with relatively little time required for steady-state algebraic solution and computation of optimization-relevant gradients. Therefore, acceleration of the pseudo-transient flowsheet simulation, as in this chapter, is highly desirable for process flowsheet optimization applications and has the largest potential for improving the computational performance of pseudo-transient flowsheet optimization.

7.3 Numerical Results

7.3.1 Simulation of Simple Example

I now revisit the Simple Example in Section 7.1 and implement the proposed strategy to solve the steady-state equations following the time-scale hierarchy. Figure 7.6 graphically shows the proposed procedure applied to this simple system where $k = 3$. As before, the algebraic equations (7.8)–(7.12) comprise $\mathbf{g}$. The standard initial conditions (7.15)–(7.17) for the state variables ($K_{i,0}$ and T_0) are employed. The system dynamics in a hierarchy of three time scales $\theta_1 = 1/\tau_1$, $\theta_2 = 1/\tau_2$, and $\theta_3 = 1/\tau_3$, correspond respectively to the dynamic *mass* balance (7.7), the *equilibrium* pseudo-transient equations (7.13), and the dynamic *heat* balance (7.14). The associated state variables are $\mathbf{x}_{d,1} = V$, $\mathbf{x}_{d,2} = K_1, ..., K_n$, and $\mathbf{x}_{d,3} = T$.

The model (7.7)–(7.14) was solved with the proposed methodology for various values of the hyper-parameters nPT and $nAlg$. Simulations were performed with $nPT = 1, 2, 3, 4, 5$. For each value of nPT, simulation was carried out with $nAlg = 5, 10, 20$. Residuals of the three components of the pseudo-transient dynamics for a subset of the simulations are shown in Figure 7.7. All simulations converged to the true solution, with an average required com-

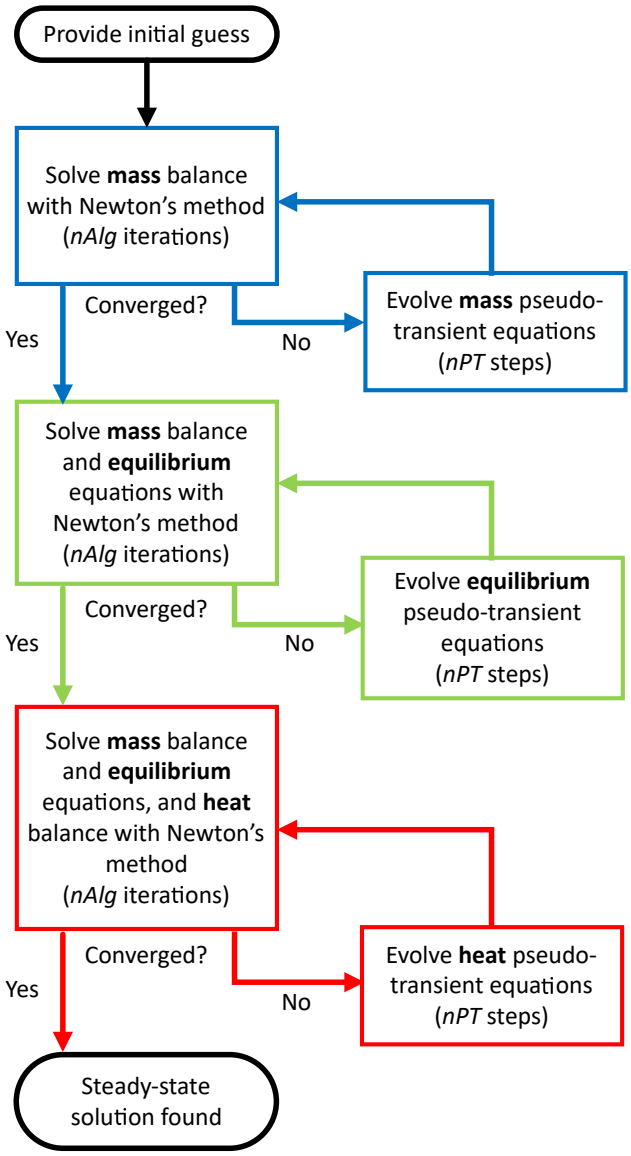

Figure 7.6: Application of the proposed quasi-steady-state method to PH flash.

putation time of 0.24s and a standard deviation of 0.10s. The computation time required for all simulations was roughly the same, as simulation with lower nPT results in fewer PT steps, but more Newton iterations (Newton method is attempted after every nPT steps). Similarly, simulation with a higher $nAlg$ can decrease the number of pseudo-transient integration steps required, as the algebraic system can converges from a worse initial guess, but may

increase the number of Newton iterations attempted. The Newton steps were carried out without step-size control, and the forward finite difference method was used to approximate derivatives of the physical properties in Jacobian evaluations.

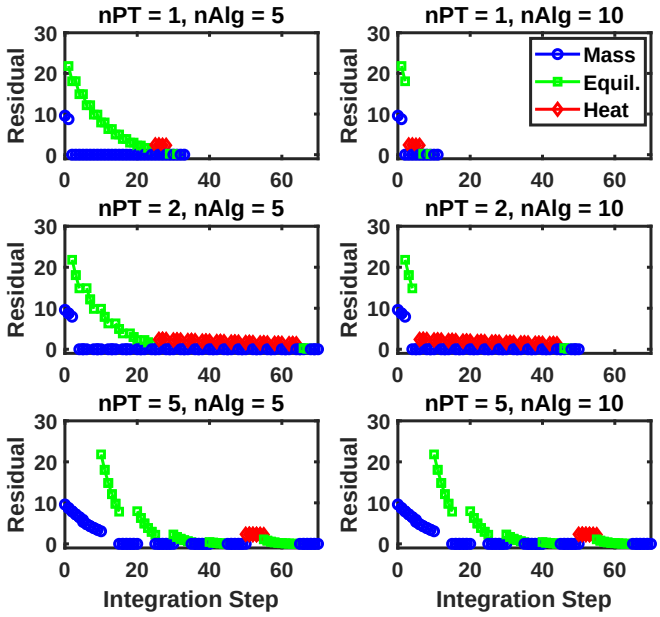

Figure 7.7: Simulation of pressure-enthalpy flash with the proposed algorithm. Various values of nPT and $nAlg$ are shown.

The system converged with a reduction in the number of pseudo-transient integration steps by several orders of magnitude, and is solved in much less computation time than either the RK4 or DASOLV integration methods. For this system, increasing $nAlg$ from 5 to 10 reduced the number of pseudo-time integration steps required, but increasing $nAlg$ from 10 to 20 did not. Simple numerical methods were used in this work to demonstrate the proposed method transparently, and I note that the simulation time could be further improved by using an implicit integration solver, as the RK4 algorithm used to perform PT integration steps is slower than the implicit DASOLV algorithm. The performance could also be improved by using a more advanced algebraic solution techniques, such as step-size control [111].

7.3.2 Simulation of a Cascade of Two-Phase Separators with Recycle

I now consider the more complex process shown in Figure 7.8.

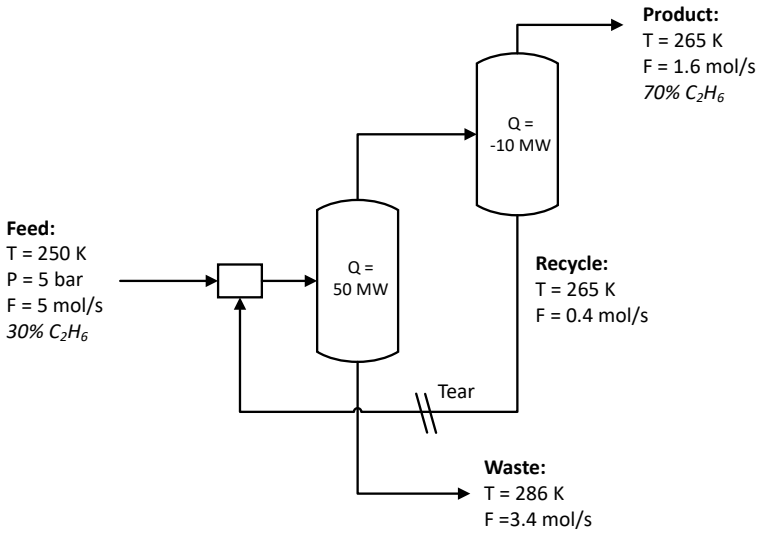

Figure 7.8: Cascade of two-phase separators with recycle.

The process structure is mathematically motivated—it is meant to demonstrate general flowsheet simulation challenges—and may not represent an actual real-world process. The vapor outlet stream of the first two-phase separator serves as the inlet stream of the second two-phase separator, and the liquid outlet of the second separator is recycled and mixed with the feed stream to recover more product. The feed stream (before the added recycle) has the same parameters as in the first example, and the first separator has the same 50MW heat input. The second separator has a a -10MW heat input (i.e., refrigeration), and a product stream containing 70% ethane is recovered as the outlet vapor stream. The conditions of all streams given by AspenPlus [9] simulation are presented in Table 7.2.

Table 7.2: Stream flow rates and mole fractions for two-stage flash flowsheet.

Stream	$Feed$	V_1	L_1	V_2	V_2
Mole Flow (mol/s)	5	2.0	3.4	1.6	0.4
ethane (%)	30	59.8	12.1	69.3	21.0
propane (%)	20	25.8	18.0	24.5	31.3
n-butane (%)	20	10.2	26.7	5.4	29.7
n-pentane (%)	25	4.0	36.0	0.9	16.9
n-hexane (%)	5	0.2	7.3	0.0	1.0

The same algebraic system of equations (7.1)–(7.6) is used to model each separator as a pressure-enthalpy flash. The following forward connectivity constraints are introduced, which represent the vapor flow from the first separator to the second:

$$M_2 z_{2,i} = V_1 y_{1,i} \tag{7.42}$$

$$H_{in,2} = H_1^V \tag{7.43}$$

The material recycle flow from the second separator back to the first represents backward connectivity and introduces algebraic loops (Figure 7.8). These backward connectivity constraints are "torn," and additional variables $\boldsymbol{z}_{tear}$, H_{tear}, and M_{tear} are defined:

$$M_1 z_{1,i} = M_{feed} z_{feed,i} + M_{tear} z_{tear,i} \tag{7.44}$$

$$\sum \boldsymbol{z}_1 = \sum \boldsymbol{z}_{tear} = 1 \tag{7.45}$$

$$H_{in,1} = H_{tear} + H_{feed} \tag{7.46}$$

$$H_{tear} = H_2^L \tag{7.47}$$

$$M_{tear} z_{tear,i} = F_{tear,i} L_2 x_{2,i} \tag{7.48}$$

The model of this integrated process contains and algebraic loop cannot be solved sequentially: the outlet conditions of the first two-phase separator are needed to simulate the second, and the outlet conditions of the second two-phase separator are needed to simulate the first. The dynamic procedure given by (3.12) is used to solve the "torn" recycle stream,

decoupling the equations via a first-order filter to (7.47)–(7.48):

$$\tau_4 \frac{dH_{tear}}{d\hat{t}} = H_2^L - H_{tear} \tag{7.49}$$

$$\tau_4 \frac{dF_{tear,i}}{d\hat{t}} = F_{2,i} - F_{tear,i} \tag{7.50}$$

where $F_{tear,i}$ is the molar flow rate of component i in the recycle stream, and $F_{2,i}$ is the same quantity for the liquid outlet of the second flash tank. H_{tear} and H_2^L are respectively the enthalpies of the recycle stream and the liquid outlet of the second flash tank. An initial guess is required for the tear stream properties in (7.49)–(7.50). A simple initial guess is to hypothesize that each separator (7.7)–(7.14) divides its inlet stream into two outlet streams of equal flow rate and composition, or $L = V = M/2$, $\boldsymbol{x} = \boldsymbol{y} = \boldsymbol{z}$. The initial guess for the tear stream flow rate was thus one quarter of the feed stream flow rate and the feed composition was used as the initial guess for the tear stream composition. The molar enthalpy of the feed stream was used as the initial guess for the tear stream molar enthalpy.

The Peng-Robinson equation of state was again used to compute the requisite physical properties. The new time constant was set as $\tau_4 = 10^3$, placing the tear dynamics in the slowest pseudo-transient time scale. This hierarchy of dynamics follows the intuitive physical motivation shown in Figure 3.2, as intra-unit (plant-wide) dynamics are typically slower than inter-unit dynamics, such as fluid-phase and thermal equilibria. Simulation of the pseudo-transient model with the same RK4 integration method required 90.4s with a tolerance of 10^{-8} and 97.8s with a tolerance of 10^{-10}. The time constants are of different orders of magnitude and allow for the dynamics in the four separate time scales to be decoupled, reaching their respective steady states sequentially. The residuals of the four pseudo-transient dynamics (mass, equilibrium, heat, tear stream) are plotted in Figure 7.9. It can be seen that the number of integration steps increases significantly compared to simulation of a single two-phase separator. The slowest dynamics are now an order of magnitude slower and

consequently require integration over a longer pseudo-time horizon to reach steady state.

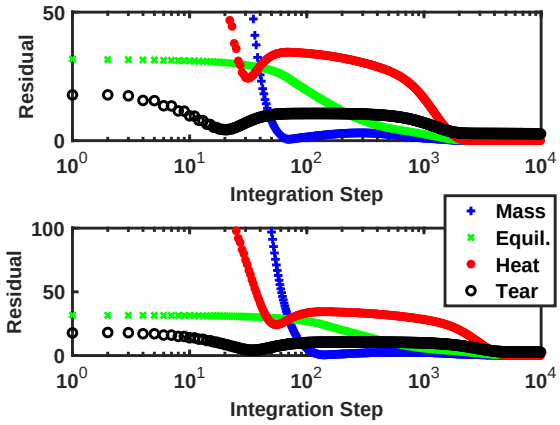

Figure 7.9: Behavior of mass, phase equilibrium, heat, and tear pseudo-transient equations during time integration. The top plot shows simulation with an RK4 integration tolerance of 10^{-8}, and the bottom plot shows simulation with a tolerance of 10^{-10}.

Simulation of the process flowsheet (Figure 7.8) was then performed with the proposed algorithm. The tear stream dynamics were treated as a fourth dynamic "layer" after the mass, equilibrium, and heat pseudo-transient equations ($k = 4$). The four time scales correspond respectively to the dynamic *mass* balance (7.7), the *equilibrium* pseudo-transient equations (7.13), the dynamic *heat* balance (7.14), the dynamic *tear* equations (7.49)–(7.50). The associated state variables are $\mathbf{x}_{d,1} = V$, $\mathbf{x}_{d,2} = K_1, ..., K_n$, $\mathbf{x}_{d,3} = T$, and $\mathbf{x}_{d,4} = [H_{tear}, F_{tear,1}, ..., F_{tear,n}]$. The remaining algebraic equations (7.8)–(7.12) for each separator and the forward connectivity equations (7.1)–(7.6), (7.42) comprise $\mathbf{g}$. The simulation was run with $nPT = 1, 2, 3, 4, 5$ and $nAlg = 1, 5, 10$, and the trajectories of the four residuals for a subset of the values of nPT and $nAlg$ are shown in Figure 7.10.

All simulations converged to the true solution, with an average computation time of 0.54s and standard deviation of 0.13s. Compared to only using PT integration, where the computation time increased by an order of magnitude from the single separator example to

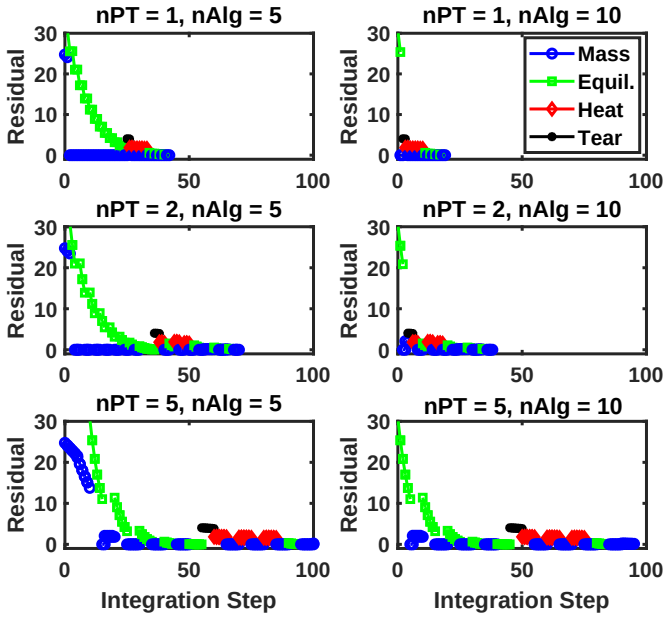

Figure 7.10: Simulation of two-stage flash process with the proposed algorithm. Various values of nPT and $nAlg$ are shown.

this more complex process, the increase in computation time for the proposed method is less dramatic. Although more PT integration steps were taken to recover the steady-state solution of the full system, the added tear stream dynamic does not have to be slower than the other systems and does not affect the integration horizon—and computation time—by order(s) of magnitude. Large-scale pseudo-transient flowsheets may include more than four layers in their hierarchy of time scales (Figure 3.2), and this reduction in simulation time may be even more beneficial to flowsheets with pseudo-transient dynamics that would require a large spectrum of time scales. Note that the number of time scales increases with the number of physical phenomena and recycle streams. This is in contrast to traditional block-based decomposition approaches [240, 282], which scale with the number of unit operations.

7.4 Summary

The underlying motivation of this chapter is that PTC should ideally be used to inform initial guesses for algebraic solvers, which typically exhibit super-linear convergence. This chapter showed that quasi-steady-state assumptions can be used to sequentially and gradually convert the reformulated dynamic equations of a pseudo-transient model to algebraic equations, allowing a gradual transition from time integration to algebraic solution of the original system. The proposed sequential algorithm further eliminates the need for user-selected time constants in pseudo-transient flowsheets, instead using a hierarchy of dynamics to converge steady-state equations in order. I demonstrated that the proposed approach performs favorably compared to conventional pseudo-transient simulation, which relies on integrating the reformulated system until steady state is reached. The developments of this chapter are general, and the techniques can be used to expedite the solution of (pseudo-transient) steady-state systems in other fields, which exhibit dynamically decoupled behavior. In addition, the techniques can be used to improve the performance of pseudo-transient models in time-relaxation-based optimization calculations, by replacing the steady-state time integration between optimization iterations.

Part II

Optimal Process Design for Variable Operation

Chapter 8

Process Design Optimization Under Uncertainty: Preliminaries[†]

8.1 Overview

Chapter 2 described various sources of uncertainty in process flowsheet models in the context of process design and optimization. Pilot testing for process designs is costly and time-consuming, and some data concerning new technologies may be unavailable or missing; in turn, this hinders the development of precise and predictive models. Furthermore, information "silos" lead to a disconnect between those who possess detailed information and those who would directly benefit from process optimization. Feedstock and product slates can change daily (or more often) [171], presenting an external source of uncertainty. Such fluctuations often cause process operation at a single fixed steady state sub-optimal or impossible. In summary, fluctuating market conditions and increasingly diverse portfolios feedstock and products are at the origin of increased uncertainty in the design and operation of chemical and energy-generation processes [15], leading to variations in both exogenous parameters (e.g. feedstream quality, product demand, utility pricing) and in endogenous parameters (e.g. reaction rate constants, efficiencies, heat transfer properties).

It is therefore critical to ensure that the resulting new designs take into account such parameter uncertainty [31, 277], a problem termed *(process) optimization under uncertainty*.

[†]The contents of this chapter are largely based on the following publication: C. Tsay, R.C. Pattison, and M. Baldea. A dynamic optimization approach to probabilistic process design under uncertainty. Ind. Eng. Chem. Res. 56(30):8606–8621, 2017. C.T. is the primary author of the manuscript.

In this chapter, I provide some background on the problem of optimization under uncertainty, as well as formulations and techniques for its solution.

8.2 Optimization Under Uncertainty

In its most general form, the problem of optimization under uncertainty (i.e., the parameters are drawn from continuous probability distributions) is infinite-dimensional, and solution approaches rely on the discretization of the stochastic variables to approximate the general problem [217]. For steady-state process design under uncertainty, these strategies fall into three major categories [203]:

1. Deterministic: the infinite-dimensional optimal design problem is approximated as a multi-scenario (multistage) optimization program, for which decomposition algorithms have been well-studied [101, 198]. A finite number of uncertain parameter values is specified *a priori*, resulting in a finite set of scenarios to be considered simultaneously as an approximation to the continuous probability distribution of the parameter.

2. Probabilistic: generally solved as two-stage stochastic program. The decision variables are divided into two sets; in the first stage, the optimal values of the design variables (e.g., equipment sizes) are selected. The second stage assumes that the uncertainty is realized, and consists of selecting the recourse (or control) variables (typically operating characteristics such as flow rates, pressures) to deal with this uncertainty. The second stage problem is itself solved using a scenario-based approach [191, 201, 217].

3. Flexibility Analysis: requires that additional information be incorporated in the optimal design problem in order to provide a plant resiliency metric. The problem is then solved typically using a robust programming formulation [100], with the aim of minimizing plant cost or maximizing profit [91].

I note that deterministic and probabilistic approaches effectively consider the optimization of the *expected* value of the objective function.

In a different vein, Blanco-Gutiérrez et al.[105, 106] proposed an interesting reformulation of the multi-scenario problem, based on converting the corresponding (likely large-scale, potentially mixed-integer nonlinear) program to a dynamic optimization, whereby scenarios are arranged chronologically in a pseudo-time domain rather than solved simultaneously. This resulted in reduced computational cost and improved convergence properties owing to reducing the number of (expensive and unreliable) initialization calculations required. Wang and Baldea extended this idea to use pseudo-random multilevel signals [277], in which the uncertain parameters are still given piecewise constant trajectories, but their levels are instead sampled from known statistical distributions.

In spite of these advances, the use of scenarios to approximate continuous probability distributions remains the fundamental approach for process design optimization under uncertainty. The resulting optimization programs are typically tractable (largely owing to decomposition strategies) when the dimensions of the underlying model are small, the set of uncertain parameters is limited, and the number of scenarios considered is low. They quickly become extremely high-dimensional and practically intractable if these conditions are not met, such as when the number of scenarios is increased to improve solution accuracy [100].

8.3 Optimizing Expected Steady-State Process Performance

8.3.1 Mathematical Approaches

Consider a general steady-state process model, which can be represented by a set of algebraic equations consisting of material and energy balances, as given by (3.1). The decision variables $\mathbf{z}$ can be partitioned to $\mathbf{d}$, the vector of design variables, and $\mathbf{u} = \mathbf{z} \setminus \mathbf{d}$,

the vector of control (recourse) variables. Replacing $\mathbf{z}$ with $[\mathbf{d}, \mathbf{u}]$ in (3.1) gives:

$$\mathbf{f}_{ss}(\mathbf{x}, \mathbf{d}, \mathbf{u}, \boldsymbol{\theta}) = 0 \tag{8.1}$$

where $\boldsymbol{\theta}$ is still a vector of of the process model parameters. In the case of parametric uncertainty, some parameter(s) in $\boldsymbol{\theta}$ is/are assumed to follow a continuous probability density function $f(\boldsymbol{\theta})$. For the case where $\boldsymbol{\theta}$ is composed of a single uncertain parameter θ_1, the expected value of some objective function $J(\mathbf{x}, \mathbf{d}, \mathbf{u}, \theta_1)$ can be expressed:

$$E[J(\mathbf{x}, \mathbf{d}, \mathbf{u}, \theta_1)] = \int J(\mathbf{x}, \mathbf{d}, \mathbf{u}, \theta_1) f(\theta_1) d\theta_1 \tag{8.2}$$

The optimal design problem can then be written as:

$$\max_{\mathbf{d}, \mathbf{u}(\theta_1)} E[J(\mathbf{x}(\theta_1), \mathbf{d}, \mathbf{u}(\theta_1), \theta_1)] \tag{8.3}$$

$$s.t. \quad \mathbf{f}_{ss}(\mathbf{x}(\theta_1), \mathbf{d}, \mathbf{u}(\theta_1), \theta_1) = 0, \forall\ \theta_1 \tag{8.4}$$

$$\mathbf{d} \in D, \mathbf{u}(\theta_1) \in U, \mathbf{x}(\theta_1) \in X \tag{8.5}$$

In lieu of discretizing the probability distribution $f(\boldsymbol{\theta})$, I will merely assume that the values of the uncertain parameters can be bounded ($\boldsymbol{\theta}^L \leq \boldsymbol{\theta} \leq \boldsymbol{\theta}^U$). This assumption naturally allows omission of values of $\boldsymbol{\theta}$ for which the probability $f(\boldsymbol{\theta})$ is negligible, or $\boldsymbol{\theta}$ values known to be impossible (i.e., outside the support of $f(\boldsymbol{\theta})$). Considering again the case of single uncertain parameter θ_1, the objective function (8.2) can be written as:

$$E[J(\mathbf{x}, \mathbf{d}, \mathbf{u}, \theta_1)] = \frac{1}{\int_{\theta_1^L}^{\theta_1^U} f(\phi) d\phi} \int_{\theta_1^L}^{\theta_1^U} J(\mathbf{x}, \mathbf{d}, \mathbf{u}, \theta) f(\theta) d\theta \tag{8.6}$$

As the integrals in (8.2) and (8.6) are generally difficult to evaluate (especially as the number of uncertain parameters increases), optimization problems are typically solved by approximating the expected value of the objective function, using a finite-dimensional expression based on discretizing the probability density function of the uncertain parameters.

Considering again the case of a single uncertain parameter θ_1, this means that n values (samples) of θ_1, i.e., $\{\theta_{1,1}, \theta_{1,2}, \ldots, \theta_{1,n}\}$ are specified in advance. Assuming these are evenly spaced (equal width discretization [86]) in the support of the probability density function, the expected value of the objective function can be estimated as [86]:

$$\hat{E}[J(\mathbf{x}, \mathbf{d}, \mathbf{u}, \theta_1)] = \frac{1}{\sum_{j=1}^{n} f(\theta_{1,j})} \sum_{i=1}^{n} f(\theta_{1,i}) J(\mathbf{x}_i, \mathbf{d}, \mathbf{u}_i, \theta_{1,i}) \tag{8.7}$$

Alternatively, an uneven sampling can be used, with, e.g., more samples closer to the mean of the distribution, along with appropriate weighting of each scenario. In either case, this multiple-scenario discretization allows the (infinite-dimensional) optimization problem (8.3)–(8.5) to be approximated as a finite-dimensional problem of the form:

$$\max_{\mathbf{d}, \mathbf{u}_1, \ldots, \mathbf{u}_n} \quad \hat{E}[J(\mathbf{x}_1, \ldots, \mathbf{x}_n, \mathbf{d}, \mathbf{u}_1, \ldots, \mathbf{u}_n, \theta_1)] \tag{8.8}$$

$$s.t. \quad \mathbf{f}_{ss}(\mathbf{x}_i, \mathbf{d}, \mathbf{u}_i, \theta_{1,i}) = 0, \forall\ i = 1, \ldots, n \tag{8.9}$$

$$\mathbf{d} \in D, \mathbf{u}_1, \ldots, \mathbf{u}_n \in U, \mathbf{x}_1, \ldots \mathbf{x}_n \in X \tag{8.10}$$

As the number of discretization points (scenarios) increases, (8.8)–(8.10) becomes a better approximation of the (infinite-dimensional) probabilistic problem (8.3)–(8.5), and the two are equivalent if an infinite number of discretization points is used [12]. However, the computational expense of solving such discretized problems is very large: assuming there are $\mathcal{K}$ uncertain parameters instead of just one and $\mathcal{Q}$ scenarios are used to discretize each parameter, a total number of $\mathcal{Q}^{\mathcal{K}}$ possible cases must be considered, a number that increases significantly with both $\mathcal{Q}$ and $\mathcal{K}$. A variant of multi-scenario optimization, sample average approximation (SAA) [156] limits the number of scenarios to be considered. Here, the total number of cases N_{SAA} is determined *a priori*, and the expected value of the objective function $J(\mathbf{x}, \mathbf{d}, \mathbf{u}, \boldsymbol{\theta})$ is determined by randomly sampling the probability density functions

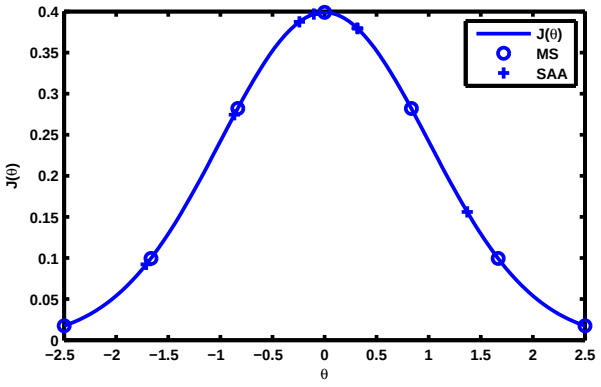

Figure 8.1: Approximating a continuous probability density function $f(\theta_1)$ using the multi-scenario (MS) and sample average approximation (SAA) approaches. Both example discretizations are generated with seven total samples.

of the uncertain parameters N_{SAA} times. A comparison of the (equal width discretization) multiple scenario and sample average approximation approaches is shown in Figure 8.1.

After generating N_{SAA} realizations of the uncertain variables by sampling their probability density function(s), $E[J(\mathbf{x}, \mathbf{d}, \mathbf{u}, \boldsymbol{\theta})]$ can be approximated as the average of the objective function values, transforming the original infinite formulation (8.3)–(8.5) to:

$$\max_{\mathbf{d},\mathbf{d}_1,\ldots,bfd_{N_{SAA}}} \quad \hat{E}[J(\mathbf{x}, \mathbf{d}, \mathbf{u}, \boldsymbol{\theta})] = \frac{1}{N_{SAA}} \sum_{i=1}^{N_{SAA}} J(\mathbf{x}_i, \mathbf{d}, \mathbf{u}_i, \boldsymbol{\theta}_i) \tag{8.11}$$

$$s.t.\ \mathbf{f}_{ss}(\mathbf{x}_i, \mathbf{d}, \mathbf{u}_i, \boldsymbol{\theta}_i) = 0, \forall\ i = 1, ..., N_{SAA} \tag{8.12}$$

$$\mathbf{d} \in D, \mathbf{u}_1, ..., \mathbf{u}_{N_{SAA}} \in U, \mathbf{x}_1, ...\mathbf{x}_{N_{SAA}} \in X \tag{8.13}$$

This formulation (8.11)–(8.13) represents a Monte Carlo-type approximation, with the accuracy of the representation also increasing with the number of sample points. Similar to the multi-scenario case, the problem formulation is equivalent to the infinite-dimensional problem (8.3)–(8.5) as N_{SAA} approaches infinity, but the computational expense of the approach increases with N_{SAA}.

8.3.2 Practical Challenges

The above considerations indicate that scenario-based approaches can become extremely computationally intensive when model accuracy (and dimensionality) and solution accuracy (in terms of the number of scenarios considered) are high, as shown in previous studies [12, 201, 303]. The application of these techniques to the optimal design of chemical processes carries an additional numerical challenge, related to solving (simulating) the process model for each scenario as required to evaluate the objective function.

Specifically, in practical situations, this entails solving a large-scale, ill-conditioned system of nonlinear equations of the form (3.1) or, equivalently, (8.1), for multiple and potentially quite disparate sets of values of $\mathbf{d}$, $\mathbf{u}$, and $\boldsymbol{\theta}$. This challenge calls for the use of robust initialization methods, such as the approach introduced in Chapter 3.

8.3.3 Parallel vs. Series Computation

In both the multi-scenario and sample average approximation approaches, a finite number of combinations of parameter values, or *scenarios*, is considered. The set of algebraic constraints corresponding to the steady-state process model (8.1) must be solved in each scenario to identify the optimal values of the operating variables and objective function. The expected value of the objective function, e.g., in (8.8) and (8.11), depends on all involved scenarios, but the scenarios themselves are independent (save for the "complicating" design variables $\mathbf{d}$) and can be solved in parallel. These observations are illustrated in Figure 8.2, and it has been recognized early on that scenario-based formulations are amenable to parallelization and/or various decomposition schemes [101].

Although the included scenarios can be solved simultaneously, the parallel approach is not always computationally favorable for process design under uncertainty [106]. Specifically, as mentioned previously, each scenario represents an independent process flowsheet and

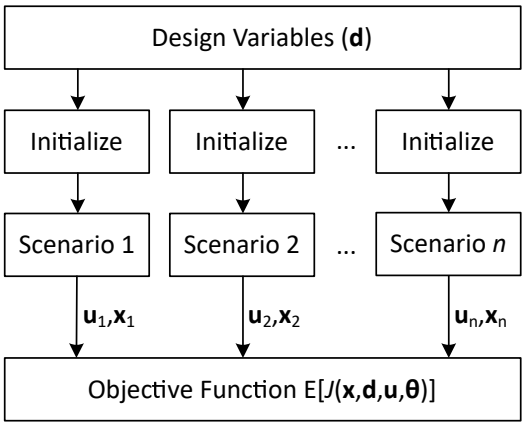

Figure 8.2: Parallel solution approach for the multi-scenario problem.

must be initialized separately (e.g. using the pseudo-transient approach of Chapter 3) for each optimization iteration, and all state variables must be stored for each scenario. Given these memory and initialization challenges, the computational requirements can become overwhelming as the number of uncertain parameters, $\mathcal{K}$, and/or the number of discretization points, $\mathcal{Q}$, for each probability density function grows.

As an alternative, the scenarios can be solved in series [105, 106], as illustrated in Figure 8.3. This approach reduces the problem size to a single process flowsheet that must be initialized, and only a single set of the state variables must be stored. The trade off is an extended computation time, as scenarios are visited sequentially.

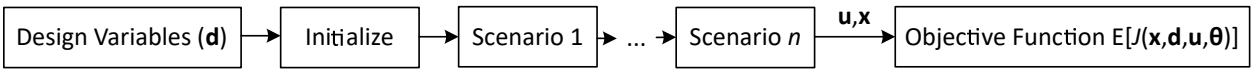

Figure 8.3: Series solution approach for the multi-scenario problem.

The series solution approach only requires a single initialization, under the assumption that an algebraic solver can reliably solve all model equations of a scenario with the initial guess being a previous scenario. However, it is difficult to guarantee convergence of any general process flowsheet.

Chapter 9

A Dynamic Optimization Approach to Probabilistic Process Design under Uncertainty[†]

In this chapter, I present a novel approach to address the "curse of dimensionality" in optimization under uncertainty. I forego the discretization of the continuous probability density functions representing the uncertain parameters and preserve the infinite-dimensional nature of the problem by converting the (steady-state) design problem into an equivalent dynamic optimization formulation, whereby the uncertain parameters are represented as a set of (continuous) disturbance variables. I present a methodology for carefully designing the trajectories of these variables (in a fictitious, *pseudo-time* domain) such that their values efficiently explore the uncertainty space. The formulation of the design optimization problem is completed by including, i) the steady-state process model as a set of path constraints and, ii) additional constraints on process variables and inputs as necessary. As in the probabilistic case, decision variables can be divided into two groups, design variables and recourse variables, with the former being time-invariant and the latter being discretized in time using control vector parametrization. The presentation in this chapter follows closely the material published in Tsay et al. (2017) [259].

[†]The contents of this chapter are largely based on the following publication: C. Tsay, R.C. Pattison, and M. Baldea. A dynamic optimization approach to probabilistic process design under uncertainty. Ind. Eng. Chem. Res. 56(30):8606–8621, 2019. C.T. is the primary author of the manuscript.

9.1 Dynamic Optimization With Continuous Parameter Trajectories

The techniques reviewed in Chapter 8 rely on the discretization of the probability density function of the uncertain parameters, $f(\boldsymbol{\theta})$, in order to convert the optimization under uncertainty problem into a finite-dimensional, scenario-based formulation that is tractable with numerical optimization solvers. In this chapter, I begin by examining the trajectories of the uncertain parameters, or "disturbance variables," as functions of pseudo-time. Returning to the series solution approach for the multi-scenario problem (Figure 8.3), I observe that this approach effectively poses the scenario-based problem as a dynamic optimization problem, with the uncertain parameters, whose values change between scenarios, treated as time-varying disturbance variables over a pseudo-time domain, $\hat{t}$. Note that dynamics in this pseudo-time domain again retain no physical meaning. An iteration of this dynamic optimization formulation is illustrated in Figure 9.1.

In practice, the step changes of the uncertain parameter θ in Figure 9.1 are difficult to simulate, since the numerical integration procedure must effectively solve the process flowsheet for a completely different set of values for θ and $\mathbf{u}$. Therefore, continuation techniques have been applied in such formulations, in order to smooth the transition from one scenario to the next [106]. The changes in θ, the uncertain parameter (disturbance) variable, between scenarios are smoothed using continuation techniques, with the parameter slowly "evolving" through pseudo-time to its value in the next scenario, e.g.,

$$\tau \frac{d\theta}{d\hat{t}} = \frac{\theta(t_{i+1}) - \theta(t_i)}{\theta(t_i)}, \ \forall \ \hat{t} \in [t_i, t_{i+1}) \tag{9.1}$$

$$\theta(\hat{t} = t_0) = \theta_0 \tag{9.2}$$

In this manner, the trajectories of uncertain parameters θ are not piecewise constant as in Figure 9.1. Rather, they are a continuous approximation of the piecewise constant

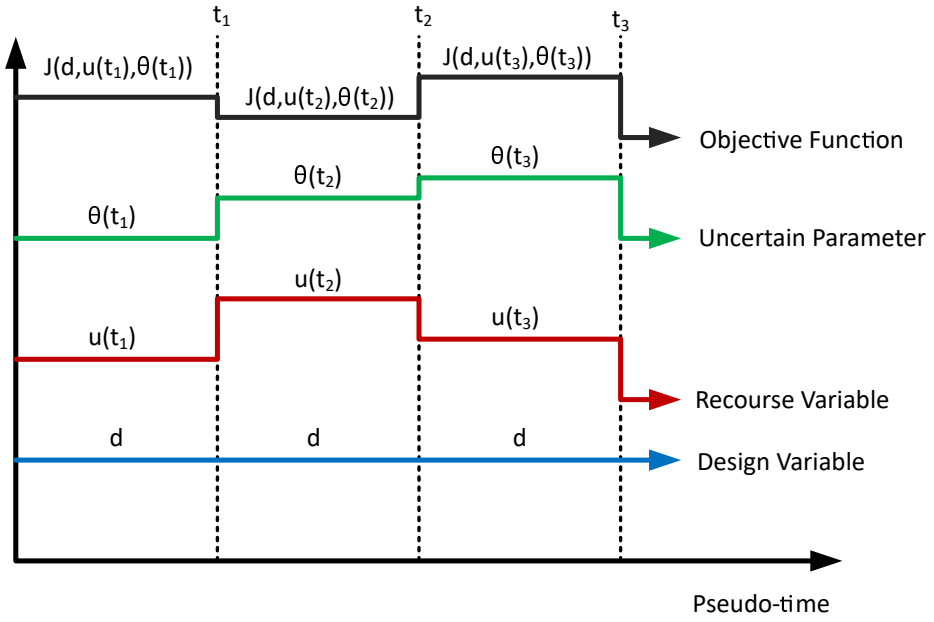

Figure 9.1: A representative iteration of the dynamic scenario-based optimization approach. The design variable d is held constant, and the recourse variable u is allowed to change for each scenario (each realization of the uncertain parameter θ). The subscripts correspond to variable values at different time intervals (where each of the latter represents a scenario).

trajectory. This formulation introduces some deviation from the multi-scenario result, as the values of θ are not always at those specified by selected scenarios. This effect can be minimized by holding the uncertain parameters constant for a large period of pseudo-time, relative to the time constant τ in (9.1), before changing to the value corresponding to the next scenario [105]. Equivalently, the "transition time" between two scnenarios (again set by the time constant τ in (9.1)) should be very short compared to the periods during which the uncertain parameters are constant. The scenarios can be ordered chronologically in a pseudo-time domain, resulting in an optimal control problem, where the control inputs (recourse variables) are piecewise constant and the states are continuous in pseudo-time.

This representation lends itself naturally to the use of a sequential dynamic optimization solution strategy, whereby the objective function and its sensitivities to decision

variables are integrated over the pseudo-time domain with the underlying process model (8.1) as algebraic path constraints. The main advantages for this series approach are computational: (i) the number of control decision variables does not increase with the number of scenarios; and (ii) only a single initial guess for the process model is required, rather than initial guesses being required to initialize each individual scenario.

Building on this approach, I consider a continuous range of values of an uncertain parameter and exploit the infinite-dimensional nature of sequential dynamic optimization. My approach is described in detail below.

9.1.1 The Case of a Single Uncertain Parameter

I consider first the case of a single uncertain parameter θ_1. Here, instead of taking discrete values that reflect a change from scenario to scenario, the value of the uncertain parameter varies continuously through pseudo-time:

$$\tau_\theta \frac{d\theta_1}{d\hat{t}} = \dot{\theta}_1(\hat{t}), \ \forall t \in [t_0, t_f) \tag{9.3}$$

$$\theta_1(\hat{t} = t_0) = \theta_{1,0} \tag{9.4}$$

where $\theta_{1,0}$ is the initial condition of θ_1, and $\dot{\theta}_1$ is the rate at which the uncertain paramter θ_1 is changed, as described below. An iteration of this dynamic optimization problem is shown in Figure 9.2.

The formulation represents a limit case of the sequential approach of Gutierrez et al. [105, 106] (Figure 9.1) as the length of time for each scenario approaches zero and the number of scenarios considered approaches infinity. Similar to the discrete-scenario case, the design variables **d** are held constant throughout the time horizon, but the recourse variables **u** are not necessarily piecewise constant. In effect, variables **u** can be represented as any (time-dependent) nonlinear function; the literature [27, 268, 269] suggests several types of such

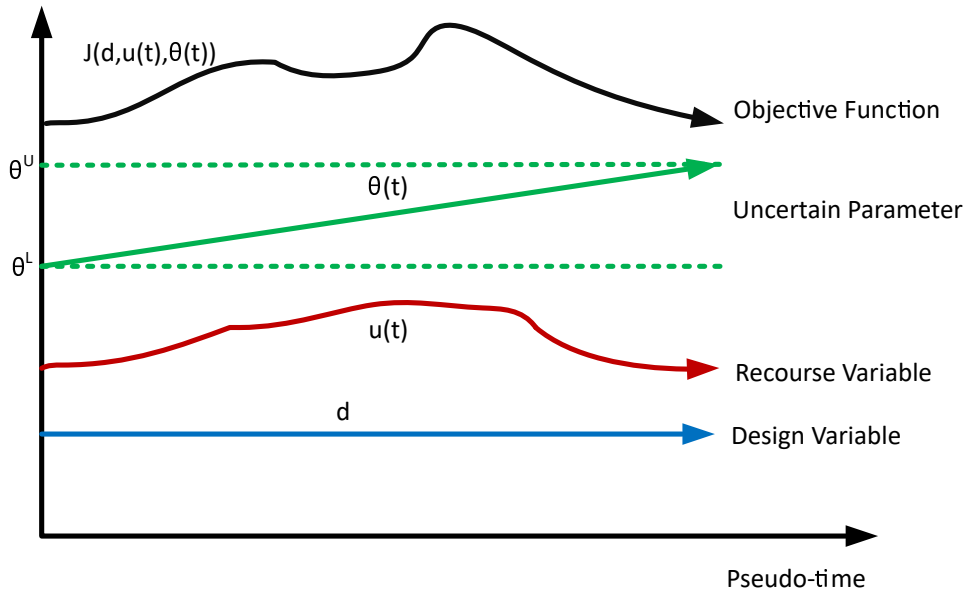

Figure 9.2: Single iteration of dynamic optimization under uncertainty using continuous representation of the uncertain parameters. The design variables are held constant, and the control variables are allowed to change through time. Also see Fig. 9.1.

functions that are useful for the purpose of performing sequential dynamic optimization. Because the uncertain parameter θ_1 varies linearly with time in this case, the abscissa in Figure 9.2 can also be considered as a scaled value of the uncertain parameter, and the representation of the control variables (control vector parametrization) determines how the optimal values of the control variables for the process vary with the uncertain parameter.

As in the scenario-based case, the objective function can be integrated through time, and the values of the integrated objective function, constraints, and their gradients can be passed to the optimization solver to determine the design variable values and control variable trajectories for the next iteration. To account for the probability distribution of the uncertain parameter, the objective function at each time point can be multiplied by the value of the probability density function of the uncertain parameter $f(\theta_1(\hat{t}))$, or the uncertain parameter trajectory $\theta_1(\hat{t})$ can be designed with the slope corresponding to $f(\theta_1(\hat{t}))$. In the former case,

the full dynamic optimization problem would be:

$$\underset{\mathbf{d},\mathbf{u}(t)}{max} \ \hat{E}[J(\mathbf{x},\mathbf{d},\mathbf{u},\theta_1)] = \int_0^{\hat{t}_f} w(\theta_1(\hat{t}))J(\mathbf{x}(\hat{t}),\mathbf{d},\mathbf{u}(\hat{t}),\theta_1(\hat{t}))d\hat{t} \tag{9.5}$$

$$s.t. \mathbf{f}_{ss}(\mathbf{x}(\hat{t}),\mathbf{d},\mathbf{u}(\hat{t}),\theta_1(\hat{t})) = 0, \forall\ \hat{t} \in [0,t_f] \tag{9.6}$$

$$\mathbf{d} \in D, \mathbf{u}(\hat{t}) \in U, \mathbf{x}(\hat{t}) \in X, \forall\ \hat{t} \in [0,t_f] \tag{9.7}$$

$$\theta_1(\hat{t}) = \frac{\theta_1^U - \theta_1^L}{t_f}\hat{t} + \theta_1^L \tag{9.8}$$

where $w(\theta_1(\hat{t}))$ is the weighting function [86] calculated from the probability density function $f(\theta_1)$ as in (8.7), where the sum over n discretization points has been replaced with an integral representing the limit as n tends to infinity. This corresponds to evaluating the equal-width discretization weights [86] over a continuous range, replacing the sum with an integral:

$$w(\theta_1) = \frac{\theta_1^U - \theta_1^L}{t_f}\frac{f(\theta_1)}{\int_{\theta_1^L}^{\theta_1^U} f(\theta)d\theta} \tag{9.9}$$

The factor $\frac{\theta_1^U-\theta_1^L}{t_f}$ is included in $w(\theta_1)$ to account for the proportionality between θ_1 and $\hat{t}$. From differentiating the equation $\theta_1(\hat{t}) = \frac{\theta_1^U-\theta_1^L}{t_f}\hat{t} + \theta_1^L$ with respect to $\hat{t}$, I obtain $d\theta_1 = \frac{\theta_1^U-\theta_1^L}{t_f}d\hat{t}$, which is then used to change the integration variable in the objective function from θ_1 to $\hat{t}$. The calculation of the objective function and state variables in this approach depends on the accurate solution of the steady-state process model (8.1) at all times during the pseudo-time integration at every iteration of the dynamic optimization. The steady-state process model can be integrated reliably for continuous changes in the disturbance variable, assuming the model equations are continuous and continuously differentiable.

Remark 9.1. Effectively, this formulation modifies the series solution approach of Gutierrez et al. [105, 106], shown in Figure 8.3, such that the series of scenarios is instead a continuous trajectory, equivalent to an infinite series of differentially-separated scenarios, as depicted in Figure 9.3. As a consequence, the proposed formulation maintains the aforementioned

computational advantages: the model only needs to be initialized once; and the number of control variables is fixed.

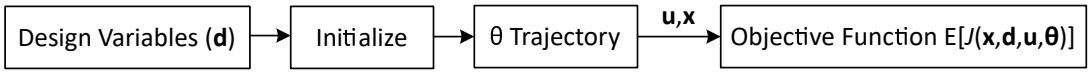

Figure 9.3: The proposed formulation is a limit case of the sequential approach [105, 106], converting a succession of scenarios representing discrete values of the uncertain parameters to a smooth, continuous trajectory.

While the concepts outlined above are completely general, from a practical perspective, I re-emphasize that (implicit) DAE solvers rely on algebraic solvers, which may experience difficulties in solving the constraint system of the proposed formulation (i.e., the algebraic process models) for some values of the uncertain parameters and/or controls. Consequently, in what follows, I propose a smoothing technique based on a pseudo-transient representation of the process model (see Chapters 3–6), which can be used to improve the numerical properties of the proposed approach. Below, I briefly review several aspects of pseudo-transient flowsheeting, as relevant to its use as a smoothing technique.

The most relevant property of a pseudo-transient flowsheet model is that the solution of the original algebraic system (8.1), or steady-state equilibrium, is recovered once the pseudo-transient system is integrated to steady state. The dynamics of the involved differential equations (3.5) can be tuned by adjusting the time constant in $\boldsymbol{\tau}$ corresponding to each equation. For the purpose of incorporating pseudo-transient models in the proposed optimization approach under uncertainty, I begin by observing that the DAE solver (time integration routine) relies on numerical integration employing a finite (pseudo-)time step $\delta\hat{t}$. In the case where the process model is presented as a set of algebraic constraints, the time step $\delta\hat{t}$ will depend exclusively on the rate of change of the parameter θ in (9.1), as captured by the time constant τ_θ. Let us now focus on the case where the process model is

in pseudo-transient form, and consider the slowest dynamic mode therein, as given in (7.24):

$$\tau_k \frac{d\mathbf{x}_{d,k}}{d\hat{t}} = \mathbf{f}_k(\mathbf{x}_d, \mathbf{x}_s, \mathbf{d}, \mathbf{u}, \boldsymbol{\theta}) \tag{9.10}$$

Since the time constant τ_k and the values of the other time constants in (7.21)–(7.24), τ_i (with $\tau_k \geq \tau_i, \quad i = 1, \ldots, k-1$, where i indexes the other, faster modes in the pseudo-transient model) can be set arbitrarily, its value can be chosen such that $\frac{\tau_k}{\tau_\theta} \ll 1$, meaning that the overall dynamic model comprising the pseudo-transient modes and the imposed dynamics of the uncertain parameters is in a standard singularly perturbed form. Thus, similar to the analysis presented in Chapter 7, the pseudo-transient modes corresponding to the process model evolve in a much faster time scale than the uncertain parameters, and can be assumed to be at a quasi-steady state at all times [145].

9.1.2 Relation to Scenario-Based Approach

The multiple-scenario optimization problem with an equal-width discretization (θ_1 is sampled at n equally spaced intervals between θ_1^L and θ_1^U, including the bounds) results in n realizations of θ_1. The scenarios $(\theta_{1,1}, \theta_{1,2}..., \theta_{1,n})$ are equally spaced, such that $\theta_{1,i} = \theta_1^L + (i-1)\Delta\theta_1$, where $\Delta\theta_1 = \theta_{1,i+1} - \theta_{1,i} = \frac{\theta_1^U - \theta_1^L}{n-1}$. Following the expressions in (8.7) and (8.8)–(8.10), the resulting multiple-scenario optimization problem from the above discretization can be written:

$$\max_{\mathbf{d}, \mathbf{u}_1, ..., \mathbf{u}_n} \frac{1}{\sum_{j=1}^{n} f(\theta_1^L + (j-1)\Delta\theta_1)} \sum_{i=1}^{n} f(\theta_1^L + (i-1)\Delta\theta_1) J(\mathbf{x}_i, \mathbf{d}, \mathbf{u}_i, \theta_1^L + (i-1)\Delta\theta_1) \tag{9.11}$$

$$s.t. \quad \mathbf{f}_{ss}(\mathbf{x}_i, \mathbf{d}, \mathbf{u}_i, \theta_1^L + (i-1)\Delta\theta_1) = 0, \forall\ i = 1, ..., n \tag{9.12}$$

$$\mathbf{d} \in D, \mathbf{u}_1, ..., \mathbf{u}_n \in U, \mathbf{x}_1, ..., \mathbf{x}_n \in X \tag{9.13}$$

As the number of scenarios n tends to infinity, $\Delta\theta_1 = \frac{\theta_1^U - \theta_1^L}{n-1} \to d\theta_1$, and the summations can be rewritten as integrals. Considering the infinite limit as $n \to \infty$, the scenario-

based formulation of the optimization problem becomes:

$$\max_{\mathbf{d},\mathbf{u}(\theta_1)} \frac{1}{\int_{\theta_1^L}^{\theta_1^U} f(\theta_1)d\theta_1} \int_{\theta_1^L}^{\theta_1^U} f(\theta_1)J(\mathbf{x}(\theta_1),\mathbf{d},\mathbf{u}(\theta_1),\theta_1)d\theta_1 \tag{9.14}$$

$$s.t. \quad \mathbf{f}_{ss}(\mathbf{x}(\theta_1),\mathbf{d},\mathbf{u}(\theta_1),\theta_1) = 0, \forall\ \theta_1 \in [\theta_1^L,\theta_1^U] \tag{9.15}$$

$$\mathbf{d} \in D, \mathbf{u}(\theta_1) \in Z, \mathbf{x}(\theta_1) \in X, \forall\ \theta_1 \in [\theta_1^L,\theta_1^U] \tag{9.16}$$

where $\mathbf{u}(\theta_1)$ is the n-dimensional vector of recourse variable decisions as $n \to \infty$ and $\mathbf{x}(\theta_1)$ is the corresponding n-dimensional vector of state variables. The proposed dynamic optimization approach for a single uncertain parameter, from combining (9.5)–(9.8) with (9.9), can be written:

$$\max_{\mathbf{d},\mathbf{u}(t)} \frac{\theta_1^U - \theta_1^L}{t_f} \frac{1}{\int_{\theta_1^L}^{\theta_1^U} f(\theta)d\theta} \int_0^{t_f} f(\theta_1(\hat{t}))J(\mathbf{x}(\hat{t}),\mathbf{d},\mathbf{u}(\hat{t}),\theta_1(\hat{t}))d\hat{t} \tag{9.17}$$

$$s.t. \quad \mathbf{f}_{ss}(\mathbf{x}(\hat{t}),\mathbf{d},\mathbf{u}(\hat{t}),\theta_1(\hat{t})) = 0, \forall\ \hat{t} \in [0,t_f] \tag{9.18}$$

$$\mathbf{d} \in D, \mathbf{u}(\hat{t}) \in U, \mathbf{x}(\hat{t}) \in X, \forall \hat{t} \in [0,t_f] \tag{9.19}$$

$$\theta_1(\hat{t}) = \frac{\theta_1^U - \theta_1^L}{t_f}\hat{t} + \theta_1^L \tag{9.20}$$

The decision variable $\mathbf{u}(\hat{t})$ is handled using a control vector parameterization technique [268], which I now consider using a piecewise-constant parameterization with m pieces. Note that, by the dynamic definition of θ_1 in Section 9.1.1, the variable of integration can be changed by using the relation $d\hat{t} = \frac{t_f}{\theta_1^U - \theta_1^L}d\theta$. The change of variable and control vector

parameterization result in the following optimization problem:

$$\max_{\mathbf{d},\mathbf{u}_1,...,\mathbf{u}_m,t_1,...,t_m} \frac{1}{\int_{\theta_1^L}^{\theta_1^U} f(\theta)d\theta} \int_{\theta_1^L}^{\theta_1^U} f(\theta)J(\mathbf{x}(\hat{t}),\mathbf{d},\mathbf{u}(\hat{t}),\theta)d\theta \tag{9.21}$$

$$s.t. \quad \mathbf{f}_{ss}(\mathbf{d},\mathbf{z}(\hat{t}),\mathbf{x}(\hat{t}),\theta_1(\hat{t})) = 0, \forall\ \hat{t} \in [0,t_f] \tag{9.22}$$

$$\mathbf{u}(t) = \quad \begin{cases} \mathbf{u}_1, \hat{t} \in [0,t_1] \\ \mathbf{u}_i, \hat{t} \in (t_{i-1},t_i]\ \forall\ i = 2,...,m \end{cases} \tag{9.23}$$

$$\mathbf{d} \in D, \mathbf{u}(\hat{t}) \in U, \mathbf{x}(\hat{t}) \in X, \forall\ \hat{t} \in [0,t_f] \tag{9.24}$$

$$\theta_1(\hat{t}) = \frac{\theta_1^U - \theta_1^L}{t_f}\hat{t} + \theta_1^L \tag{9.25}$$

In the limit as $m \to \infty$, then $t_i - t_{i-1} \to d\hat{t}\ \forall\ i = 1,...,m$, and $\mathbf{u}(\hat{t})$ becomes an infinite-dimensional control vector. Rewriting the optimization problem with this infinite limit, the general dynamic optimization formulation (9.17)–(9.20) is recovered with a change of integration variable:

$$\max_{\mathbf{d},\mathbf{u}(\hat{t})} \frac{1}{\int_{\theta_1^L}^{\theta_1^U} f(\theta)d\theta} \int_{\theta_1^L}^{\theta_1^U} f(\theta)J(\mathbf{x}(\hat{t}),\mathbf{d},\mathbf{u}(\hat{t}),\theta)d\theta \tag{9.26}$$

$$s.t. \quad \mathbf{f}_{ss}(\mathbf{x}(\hat{t}),\mathbf{d},\mathbf{u}(\hat{t}),\theta_1(\hat{t})) = 0, \forall\ \hat{t} \in [0,t_f] \tag{9.27}$$

$$\mathbf{d} \in D, \mathbf{u}(\hat{t}) \in U, \mathbf{x}(\hat{t}) \in X, \forall\ \hat{t} \in [0,t_f] \tag{9.28}$$

$$\theta_1(\hat{t}) = \frac{\theta_1^U - \theta_1^L}{t_f}\hat{t} + \theta_1^L \tag{9.29}$$

Therefore, as $m \to \infty$, the proposed dynamic approach (9.26)–(9.29) becomes the same optimization problem as the multiple scenario approach (9.14)–(9.16) with a bounded uncertain parameter for $n \to \infty$ scenarios.

Remark 9.2. This analysis does not capture the rate at which the scenario-based approach (9.11)–(9.13) converges to the dynamic optimization approach (9.21)–(9.25) with increasing number of scenarios n; rather, it is emphasized that the dynamic optimization approach (with infinite-dimensional control vector parameterization) matches the infinite-scenario limit of the scenario-based approach.

Remark 9.3. For practical implementation, the problem cannot be computationally formulated using an infinite-dimensional control vector parameterization. I propose to solve the optimization problem using a finite-dimensional control vector parameterization with progressively larger values of m. Once the change in objective function between two different values of m drops below a pre-defined threshold, I assume that the solution found with the smaller of the two values of m accurately approximates the solution as $m \to \infty$.

9.1.3 The Case of Multiple Uncertain Parameters

The description of the single uncertain parameter case above suggests that the proposed approach is, in effect, a means for exploring the one-dimensional uncertainty space by "scanning" it with the value of the uncertain parameter. Below, I extend this idea to a multi-dimensional uncertainty space. To this end, I consider a "sawtooth" exploration strategy, shown in Figure 9.4, wherein each uncertain parameter "sweeps" its full range a pre-determined finite number of times in the length of pseudo-time it takes for the next (slower) uncertain variable to explore its full range.

This effectively assigns the dynamics of the set of uncertain parameters to a hierarchy of time scales, with the search of the uncertainty space being more complete as the difference between time scales grows and the number of intersections between trajectories increases. Because this formulation ensures that for each uncertain parameter the trajectory spends an even amount of pseudo-time at all values, the expected value of the design objective can be approximated by simply integrating the probability-weighted objective function, as in (9.5)–(9.8). The resulting optimization problem has a similar form to (9.5)–(9.8):

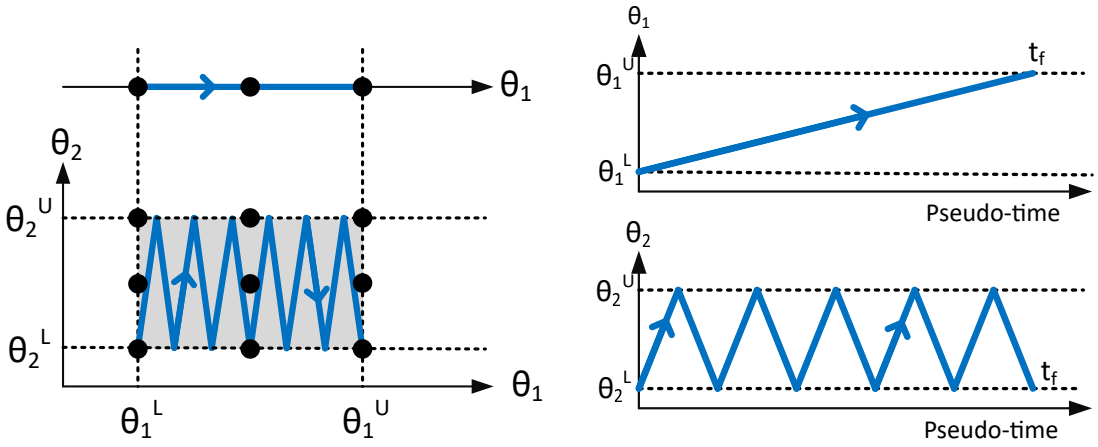

Figure 9.4: Exploration of the multivariate uncertain parameter space. Left: the black dots represent a multi-scenario discretization and the blue lines represent dynamic optimization trajectories. In a single dimension, the entire search space can be explored by the dynamic optimization formulation. In higher dimensions (the shaded area) only a finite number of paths can be explored. Right: trajectories for a sawtooth exploration in the case of two uncertain parameters (left, bottom).

$$\max_{\mathbf{d},\mathbf{z}(\hat{t})} \ \hat{E}[J(\mathbf{x},\mathbf{d},\mathbf{u},\boldsymbol{\theta})] = \int_0^{t_f} w(\boldsymbol{\theta}(\hat{t}))J(\mathbf{x}(\hat{t}),\mathbf{d},\mathbf{u}(\hat{t}),\boldsymbol{\theta}(\hat{t}))d\hat{t} \tag{9.30}$$

$$s.t. \quad \mathbf{f}_{ss}(\mathbf{x}(\hat{t}),\mathbf{x},\mathbf{u}(\hat{t}),\boldsymbol{\theta}(\hat{t})) = 0, \forall\ \hat{t} \in [0,t_f] \tag{9.31}$$

$$\mathbf{d} \in D, \mathbf{u}(\hat{t}) \in U, \mathbf{x}(\hat{t}) \in X, \forall\ \hat{t} \in [0,t_f] \tag{9.32}$$

$$\theta_1(\hat{t}) = \frac{\theta_1^U - \theta_1^L}{t_f}\hat{t} + \theta_1^L \tag{9.33}$$

$$\theta_2(\hat{t}) = \begin{cases} n_2\frac{\theta_2^U-\theta_2^L}{t_f}\hat{t} + \theta_2^L, \hat{t} \in [2\frac{t_f}{n_2}j, 2\frac{t_f}{n_2}j+1) \\ n_2\frac{\theta_2^L-\theta_2^U}{t_f}\hat{t} + \theta_2^U, \hat{t} \in [2\frac{t_f}{n_2}j-1, 2\frac{t_f}{n_2}j) \end{cases}, \ n_2 \in I, \ j = 0,1,...,n_2 \tag{9.34}$$

$$\vdots \quad \vdots \tag{9.35}$$

$$\theta_k(\hat{t}) = \begin{cases} n_k\frac{\theta_k^U-\theta_k^L}{t_f}\hat{t}, \hat{t} \in [2\frac{t_f}{n_k}j, 2\frac{t_f}{n_k}j+1) \\ n_k\frac{\theta_k^L-\theta_k^U}{t_f}\hat{t}, \hat{t} \in [2\frac{t_f}{n_k}j-1, 2\frac{t_f}{n_k}j) \end{cases}, \ n_k \in I, \ j = 0,1,...,n_k \tag{9.36}$$

where n_i is the number of times the i^{th} uncertain parameter explores its full range of uncertainty in the time it takes the slowest uncertain parameter to explore its full range once. The

fastest dynamics for the uncertain parameters corresponds to the parameter with largest n_k, and the smallest time constant (9.1) is scaled to $\frac{\tau}{n_k}$. A sample trajectory for exploring a three-dimensional parameter uncertainty space is shown in Figure 9.5.

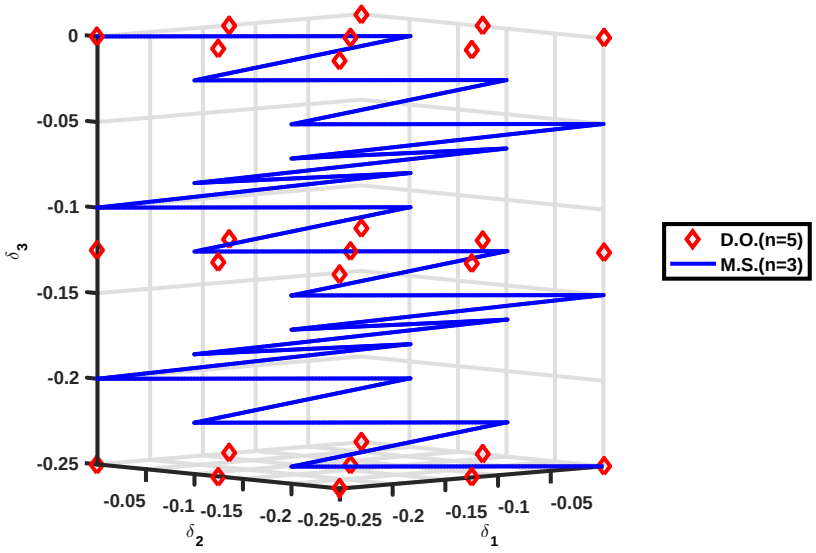

Figure 9.5: A comparison of a sample "sawtooth" trajectory (n_2=n_3=5 as described by (9.30)–(9.36)) with a multi-scenario approximation (each uncertain parameter is discretized with 3 points) for exploring a three-dimensional parameter uncertainty space. Abbreviations: D.O., dynamic optimization; M.S., multiple scenario.

Similar to the multi-scenario optimization problems, the dynamic optimization approach also lends itself to parallelization. Instead of multiple scenarios being considered and solved simultaneously, multiple trajectories can be solved simultaneously and the objective function computed from the results. As shown in Figure 9.6, this parallelization approach is essentially considering "multiple scenarios," where each scenario is a trajectory through the uncertainty space. Although each additional scenario would increase the number of variables in the problem, additional scenarios would not require a separate initialization of the flowsheet, rather the number of initializations required increases only with the number of different starting points for all the trajectories combined, as shown in Figures 9.6 and 9.7.

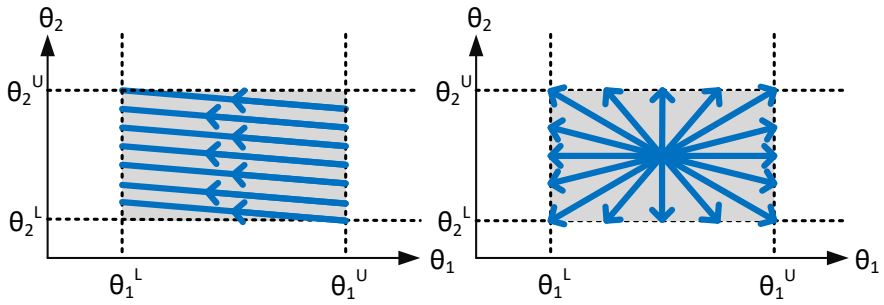

Figure 9.6: Parallelization of the dynamic optimization approach. Multiple trajectories are considered in each optimization iteration. A multiple initialization (left) and a single initialization (right) formulation are shown.

As the dimensionality of the parameter uncertainty space increases, the length of pseudo-time integration required to explore a reasonable amount of the uncertainty space with a single trajectory can grow very large. Using multiple trajectories as shown in Figure 9.7 allows the length of pseudo-time integration required to explore the parameter uncertainty space to be decreased, at the cost of increasing the memory required to store all of the threads being simultaneously integrated (similar to storing multiple scenarios).

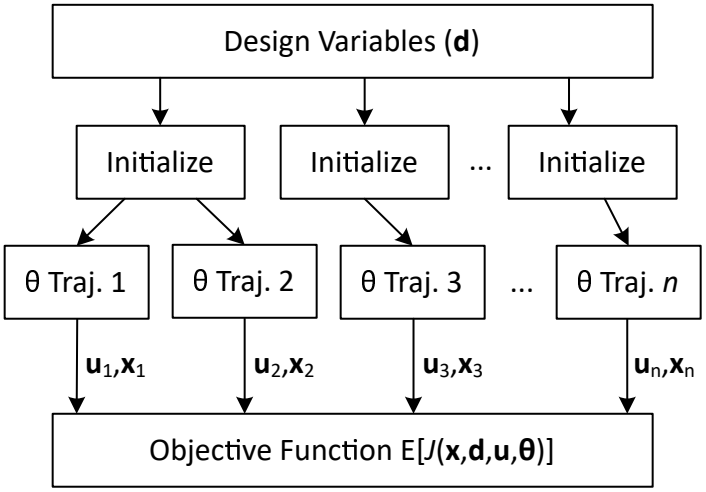

Figure 9.7: The dynamic optimization formulation parallelized to multiple trajectories.

9.1.4 Problem Formulation Considerations

Most design optimization problems include constraints, limiting the feasibility of potential designs. Since the objective function is the integral of the running process steady-state objective function, only the final value (terminal cost) is important. Equality constraints can simply be included in the process algebraic model (8.1), and inequality constraints are converted to dynamic endpoint constraints by integrating the constraint violation. The constraints could also be treated as path constraints (similar to converting the objective function from Mayer to Bolza form). For a general inequality constraint of the form:

$$h(\mathbf{x}, \mathbf{d}, \mathbf{u}, \boldsymbol{\theta}) \leq 0 \tag{9.37}$$

An equivalent dynamic endpoint constraint can be written:

$$\bar{h}(\mathbf{x}, \mathbf{d}, \mathbf{u}, \boldsymbol{\theta})|_{\hat{t}=t_f} \leq 0 \tag{9.38}$$

$$\frac{d\bar{h}}{d\hat{t}}(\mathbf{x}, \mathbf{d}, \mathbf{u}, \boldsymbol{\theta}) = \max(0, h(\mathbf{x}, \mathbf{d}, \mathbf{u}, \boldsymbol{\theta})) \tag{9.39}$$

$$\bar{h}(\mathbf{x}, \mathbf{d}, \mathbf{u}, \boldsymbol{\theta})|_{\hat{t}=0} = 0 \tag{9.40}$$

A second consideration in formulating the design under uncertainty problem as a dynamic optimization is the case of discontinuous regions of feasibility for the control variables. As a simple example, consider the design of a heater with a liquid feed and for which the outlet must be single phase, where the heat duty is a recourse (control) variable. Then, there exists a discontinuity in the feasible region for the control variable, between the heat duties required to heat the stream to its dew point and to its bubble point. Although the dynamic optimization formulation requires that control variable trajectories be continuous, discontinuities in the regions of feasibility can be addressed using parallel trajectories, as mentioned previously. As an example, suppose that a process has a control variable u_1, such

that $u_1 \in U_1 \cup U_2$. This problem can be formulated by solving two separate instances of the continuous dynamic optimization problem (9.5)–(9.8) or (9.30)–(9.36) in parallel (one instance with $u_1 \in U_1$ and the other with $u_1 \in U_2$) and a combined objective function that integrates the maximum of the objective functions found in either region U_1 or U_2. A discontinuity could also exist in the values of the uncertain parameters, such as a process designed to operate with either of two catalysts; this type of discontinuity could be similarly addressed with parallel disturbance trajectories.

9.2 Example: Design of the Williams-Otto Process

9.2.1 Process Description and Nominal Case Optimization

To compare the proposed approach against a multi-scenario formulation, I consider design under uncertainty for the simple Williams-Otto process [29, 215, 285]. All computations in this chapter are carried out in gPROMS [204] on an Intel Core i7 processor at 3.40 MHz and 16 GB of RAM, and sequential quadratic programming (NLPSQP) is used to solve all optimizations. The process flowsheet is shown in Figure 9.8.

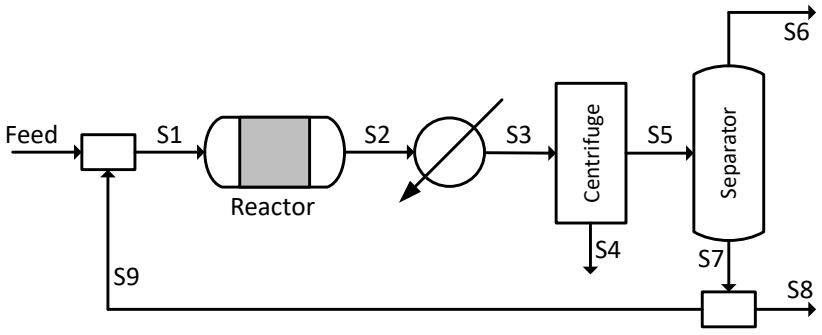

Figure 9.8: Diagram of the Williams-Otto process (based on [285]).

The Williams-Otto process is continuous and converts feed components A and B to a product P. The following three reactions occur in the reactor (modeled as a CSTR):

$$\begin{aligned} A + B &\xrightarrow{k_1} C \\ C + B &\xrightarrow{k_2} P + E \\ P + C &\xrightarrow{k_3} G \end{aligned} \tag{9.41}$$

where C is an intermediate component, E is a byproduct, and G is a waste product. The process feed stream contains only components A and B, and is mixed with a recycle stream (S9) and fed to the CSTR. The effluent of the reactor (S2) is cooled in a heat exchanger and sent to a centrifuge which separates waste product G (S4). The components of the stream with waste product removed (S5) are then separated to remove product P overhead (S6); however, due to the presence of an azeotrope, some of the product (10 wt. % of component E) is retained in the bottoms (S7). The bottoms is split into a purge stream, S8, and the recycle stream, S9. The objective of the design optimization problem is to maximize the return on investment (ROI), which is calculated [29]:

$$\text{ROI} = 100(2207F_{S6} + 50F_{S8} - 168Feed_A - 252Feed_B - 2.22F_{S2} - 84F_{S4} - 60V\rho)/(600V\rho) \tag{9.42}$$

where V is the volume of the reactor and ρ is the density of the components inside the reactor. The reaction rate constants are calculated using the Arrhenius equations:

$$k_1 = k_{1,0}\exp(-120/T) \tag{9.43}$$

$$k_2 = k_{2,0}\exp(-150/T) \tag{9.44}$$

$$k_3 = k_{3,0}\exp(-200/T) \tag{9.45}$$

where T is the temperature inside the reactor. The values for the model parameters are $\rho = 50$, $k_{1,0} = 5.9755 \times 10^9$, $k_{2,0} = 2.5962 \times 10^{12}$, and $k_{3,0} = 9.6263 \times 10^{15}$. Following the

formulation given by Biegler [29], the volume and mass flows are scaled by a factor of 10^3, and temperature is scaled by a factor of 10^2. The recycle stream is represented using a dynamic tearing method (3.12), and the steady-state time relaxation method was used to optimize the flowsheet. Note that, although this flowsheet can be solved algebraically relatively easily, in general flowsheets require an initialization method, and this approach provides more accurate estimates of practical simulation times. The lone constraint implemented is that the flow rate of product stream (S6) be less than or equal to 4.763 $\frac{lb}{1000h}$. The suggested initial point given by Biegler [29] was used, and the optimal point found is presented in Table 9.1. The deterministic optimization was solved in 6.6s, requiring a peak memory usage of 12.3MB. The recycle ratio RR is defined as the flow of stream S9 divided by the flow of the bottoms product S7.

Table 9.1: Optimized Williams-Otto process.

Variable	Lower	Optimal	Upper
Reactor Volume (V)	0.03	0.03	0.10
Reactor Temperature (T)	5.8	6.74	6.8
Product Flow (S6)	0	4.720	4.763
Recycle Ratio (RR)	-	0.90	-
Feed A	0	13.4	-
Feed B	0	30.5	-
ROI	-	121.1	-

For this case study, uncertainties are considered for the pre-exponential factors of the rate expressions. Since the reactor volume reaches its lower bound in the optimized nominal case (Table 9.1), I only consider cases in which the pre-exponential factors are decreased (i.e. the reactions are slower than expected), where I would expect the reactor size would have to be increased at the optimal point to achieve the same conversion. To model this uncertainty, variables δ_1, δ_2, and δ_3 are introduced such that $k_{i,0} = (1 + \delta_i)k_{i,0}^{nominal}$. The parameters are equal to their nominal values when the corresponding δ is zero.

9.2.2 Single Uncertain Parameter

I assume that the feed flow rate of A (the design basis) and the reactor volume V are design variables, invariant under uncertainty. The feed flow rate of B (feed ratio), reactor temperature T, and recycle ratio RR are control variables allowed to change with the uncertain parameters during optimization. The optimal values of the design variables for uncertainty in δ_1 were determined using a multi-scenario approach with varying n (discretization points were evenly spaced between δ =-0.5 and δ =0) and using the dynamic optimization approach, and the expected values of the objective function (E[ROI]) at the optimal points are shown in Figure 9.9.

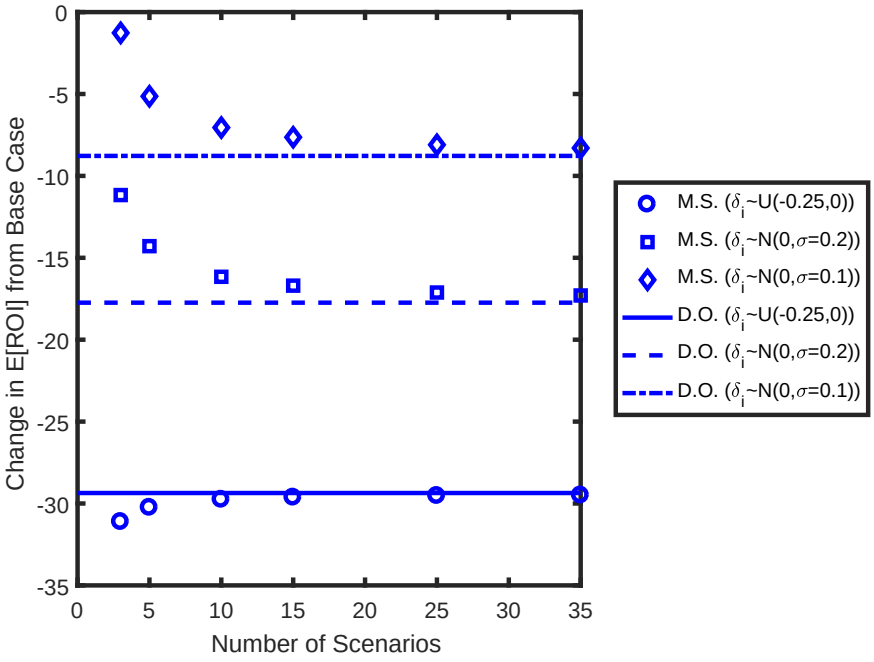

Figure 9.9: The optimal objective function found for a single uncertain parameter. M.S. stands for multiple scenario and D.O. stands for dynamic optimization.

As expected, the decrease in the objective function from the base case is larger for flatter probability distributions, as the parameter δ is more likely to deviate further from its

nominal value. Notably, for all three probability distributions, the expected values of the objective function seem to approach the value found by the dynamic optimization as the number of scenarios increases, sugggesting the expected values predicted by the dynamic optimization are the true expected values.

9.2.3 Multiple Uncertain Parameters

For the cases of uncertainties in two of the three pre-exponential constants and in three of the pre-exponential constants, the "sawtooth" approach (9.30)–(9.36) was used to define the trajectories of the uncertain parameters. The dynamic optimizations were solved both with and without smoothing the algebraic process constraints using pseudo-transient models for a comparison of the required solution times and accuracy. The dynamic optimization was solved with the dynamics of the second uncertain parameter disturbance five times and ten times as fast as the dynamics of the first uncertain parameter ($n_2 = 5, 10$ in (9.30)–(9.36)), with both formulations resulting in the same optimal point. A comparison of the optimal objective function (E[ROI]) values between a dynamic optimization formulation and a multiple scenario formulation is shown in Figure 9.10.

The dynamic optimization formulation and the multiple scenario discretization approximate the two-dimensional uncertainty space in distinct ways. The multiple scenario discretization relies on a fixed number of cases (equal to n^2), while the dynamic optimization consists of a continuous trajectory through the uncertainty space. The optimal values for the operating variables are formulated similarly to the objective function for each approach, giving the optimal operation conditions for the designed process at various values of the uncertain parameters. For values excluded from the design optimization (i.e. the uncertain parameter values do not fall exactly on a scenario or on the dynamic optimization trajectory), the operating variables can be interpolated from the optimized values or

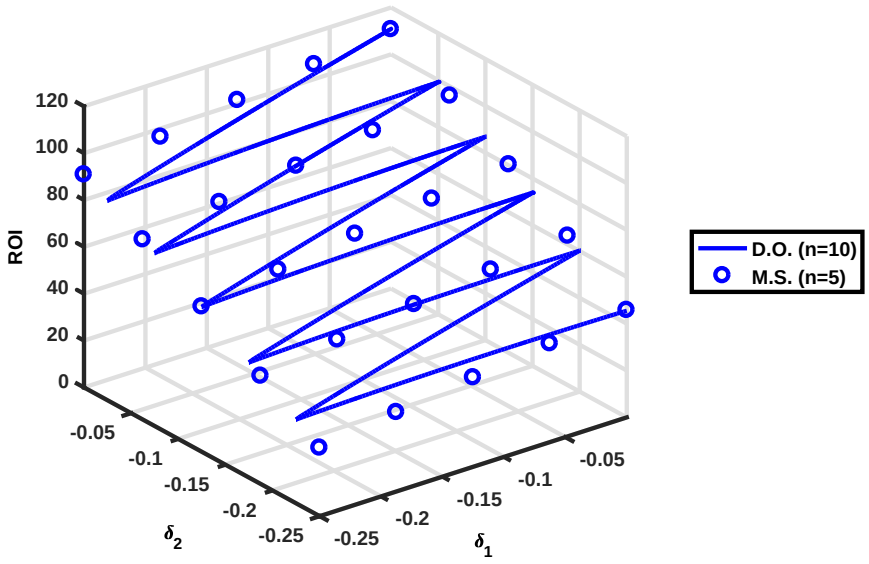

Figure 9.10: Optimized ROI for the Williams-Otto process with two uncertain parameters.

determined from a new optimization procedure.

The expected values of the objective function (ROI) for various discretization levels and various distributions for two uncertain parameters are presented in Figure 9.11. As observed earlier, the expected values of the objective function (E[ROI]) converge to the values predicted by the dynamic optimization formulation as the number of scenarios increases. With two uncertain parameters, the entire uncertainty space cannot be covered with a finite continuous trajectory ($\hat{t}$ is bounded), and thus the dynamic optimization formulation still represents an approximation of the uncertainty space; however, the optimizations with both 5 and 10 "sawtooths" reached the same optimal point, suggesting the approximation error is very small, which is confirmed by the multiple-scenario optimization results with higher numbers of scenarios.

I apply the same "sawtooth" disturbance variable formulation (9.30)–(9.36) to explore the uncertainty space for three uncertain parameters, with $n_2 = n_3 = n = 5$. Using the

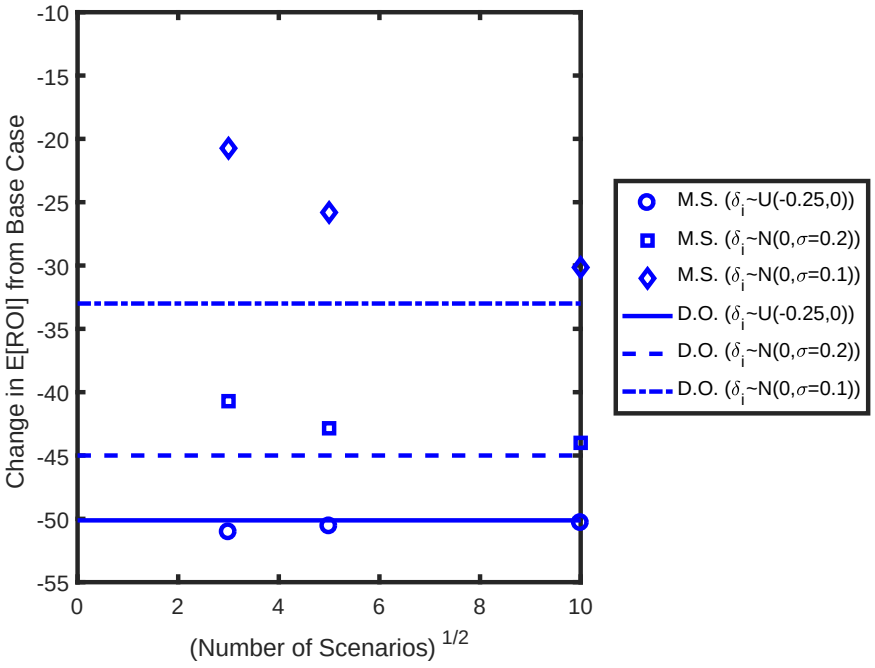

Figure 9.11: The optimal objective function found for two uncertain parameters. M.S. stands for multiple scenario and D.O. stands for dynamic optimization.

dynamic optimization with three disturbance parameter trajectories increases the pseudo-time span considerably. As mentioned previously, this could potentially be alleviated with the use of a hybrid approach, running multiple scenarios of continuous trajectories and balancing the increase in number of variables and decrease in time horizon t_f for each additional thread. The expected ROI for various discretization levels and various distributions for three uncertain parameters are presented in Figure 9.12.

Unlike the previous results, the expected return on investment in the multiple scenario simulations does not seem to approach the value found in the dynamic optimization approach as n increases; however, I attempted to test the multiple scenario formulation with $n = 7$, and the optimization calculations were not completed even after 400 hours. The dynamic optimization formulation with a single trajectory can "search" less of the parameter

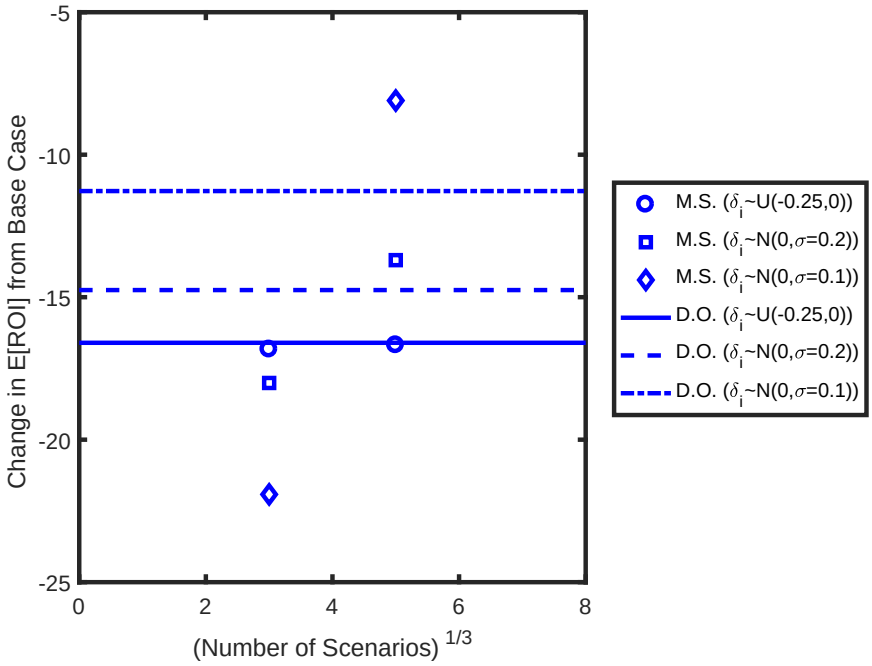

Figure 9.12: The optimal objective function found for three uncertain parameters. M.S. stands for multiple scenario and D.O. stands for dynamic optimization.

uncertainty space as the number of dimensions in the uncertainty space (number of uncertain parameters) increases, but the search space could also be explored by running multiple trajectories in parallel. However, the solutions obtained using the proposed dynamic optimization approach are very similar to scenario-based cases (especially for lower dimensional parameter uncertainty spaces), while offering massive, order-of-magnitude reductions in both CPU time and memory use as shown in Figures 9.13 and 9.14.

While, as shown in Figure 9.14 the memory usage for the multiple scenarios increases very quickly with the level of discretization, especially with multiple uncertain parameters, the peak memory usage for the dynamic optimization problems was 13.8 MB, 14.6 MB, and 17.4 MB for a single uncertain parameter, two uncertain parameters, and three uncertain parameters, respectively. Compared to only discretizing each uncertain parameter into three

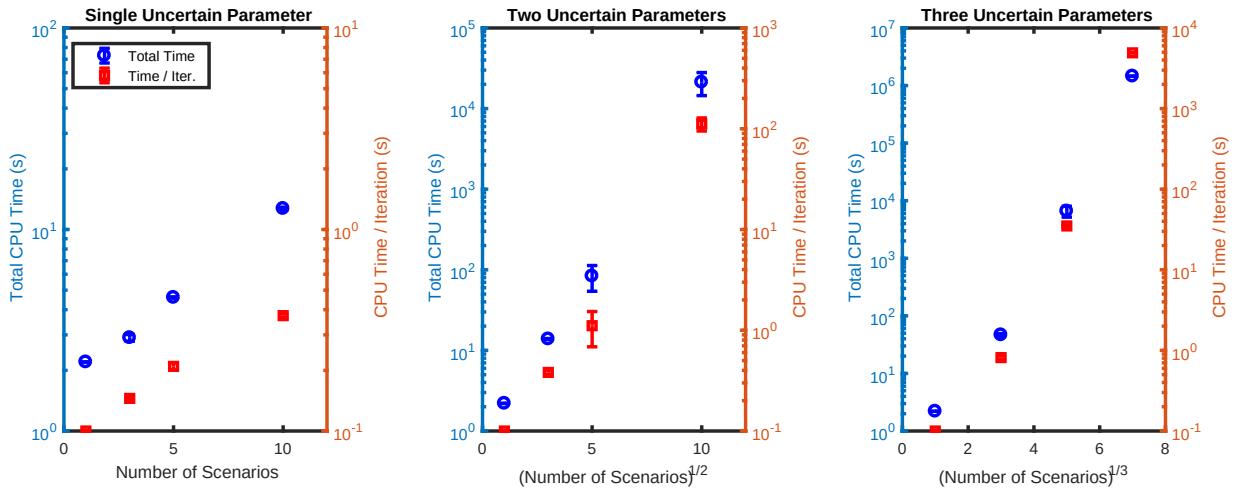

Figure 9.13: The CPU times required for the multiple scenario optimizations in this case study. The data points represent the average for all three probability distributions, and the bars represent the standard deviation.

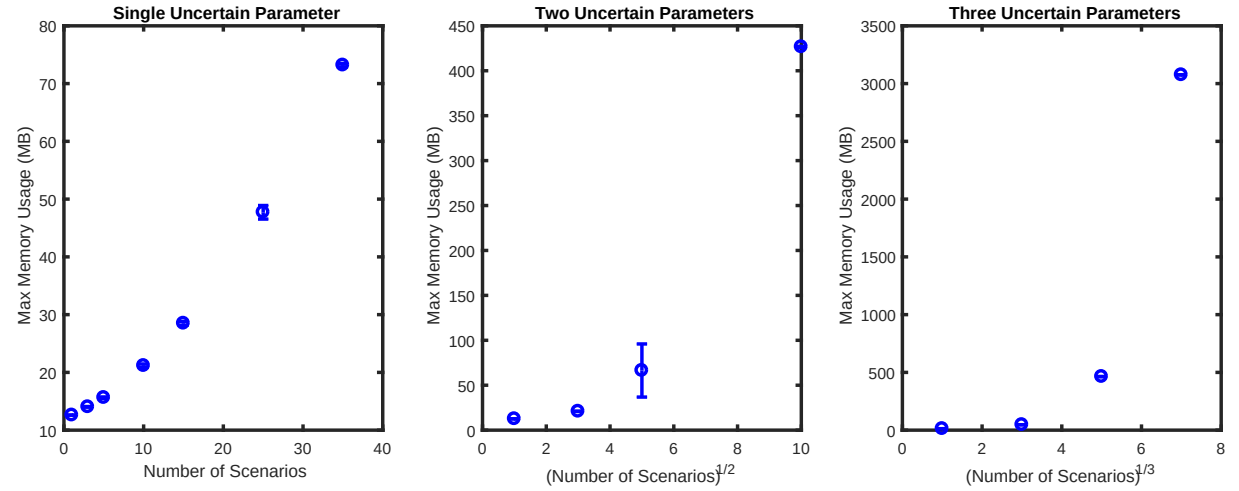

Figure 9.14: The peak memory usage required for the multiple scenario optimizations in this case study. The data points represent the average for all three probability distributions, and the bars represent the standard deviation.

samples/scenarios, the dynamic optimization approach for this case study uses about 1% less memory for a single uncertain parameter, 33% less memory for two uncertain parameters, and 66% for three uncertain parameters, emphasizing the computational benefits of the

proposed approach, especially as the dimensionality of the uncertainty space increases.

The dynamic optimizations for a single uncertain parameter were solved in 26.3 ± 9.9 s of CPU time (average ± standard deviation), equivalent to 1.4 ± 0.7 s per iteration. The dynamic optimizations for two uncertain parameters were solved in 98.7 ± 11.0 s of CPU time for $n_2 = 10$ and 43.0 ± 19.6 s of CPU time for $n_2 = 5$, equivalent to 1.9 ± 0.9 s and 1.1 ± 0.6 s per iteration respectively. The dynamic optimizations for three uncertain parameters took considerably longer, being solved in 900.1 ± 596.5 s, or 5.7 ± 3.0 s per iteration. Although some of this required CPU time, which is still considerably lower than in the multiple scenario cases (Figure 9.13), could be avoided with a better initialization procedure, as mentioned earlier, parallel trajectories could be run in each iteration, reducing the length of trajectories required to explore the parameter uncertainty space by each individual thread.

9.3 Summary

This chapter presented a novel framework for solving process design optimization problems under uncertainty, based on reformulation as a dynamic optimization problem. The uncertain parameters are represented as continuous dynamic disturbance variables in pseudo-time. The parameter uncertainty space can thus be explored efficiently with a continuous pseudo-time trajectory rather than via a finite number of samples as is the case in conventional scenario-based approaches for optimization under uncertainty. I demonstrated through two case studies that the accuracy of the dynamic optimization approach (measured in terms of the expected value of the design objective function) is very close to the values to which the scenario-based calculations converge as the number of scenarios is increased. Moreover, the proposed approach offers massive, order-of-magnitude reductions in computational effort (both CPU time and memory use), particularly in the case when the number of uncertain parameters and size of the optimization problem increase.

Chapter 10

Optimization of Reaction/Separation/Recycle Processes for Variable Operation[†]

In this chapter, I examine the optimal design of three practically relevant reaction/separation/recycle processes for variable operating points using the dynamic optimization formulation presented in Chapter 9. The three case studies comprise: (Case Study 1) the design of an ammonia synthesis plant for variable reaction rate, (Case Study 2) a revisit to the design of the dimethyl ether production process considered in Chapter 4 for variable catalyst performance, and (Case Study 3) the design of a dividing-wall distillation column for variable vapor split fraction. These case studies demonstrate the design changes that arise from considering variable operating points, revealing the importance of considering key *endogenous* parametric uncertainties at the process design stage. Furthermore, the computational requirements highlight the advantages of the proposed approach for optimization under uncertainty. The presentation of Case Study 2 in this chapter follows closely the material published in Tsay et al. (2017) [259].

[†]The contents of this chapter are largely based on the following publication and presentation (C.T. is the primary author of both):
C. Tsay, R.C. Pattison, and M. Baldea. A dynamic optimization approach to probabilistic process design under uncertainty. Ind. Eng. Chem. Res. 56(30):8606–8621, 2019.
C. Tsay, R.C. Pattison, and M. Baldea. Design of dividing-wall columns accounting for vapor split uncertainty. International Process Intensification Congress (IPIC1), Barcelona, Spain, Oct 2017.

10.1 Case Study 1: Design of Ammonia Synthesis Plant

A basic ammonia synthesis process is described in Chapter 7 of Nonlinear Programming by Biegler [29] as a prototypical steady state process design problem. The process is continuous and converts hydrogen and nitrogen feed to ammonia product by the following reaction:

$$N_2 + 3H_2 \rightarrow 2NH_3 \tag{10.1}$$

The reaction occurs at high temperature and pressure in catalytic reactors. In this case study, I consider the base case process [29] with an additional reactor to allow for more design freedom, and the flowsheet is shown in Figure 10.1.

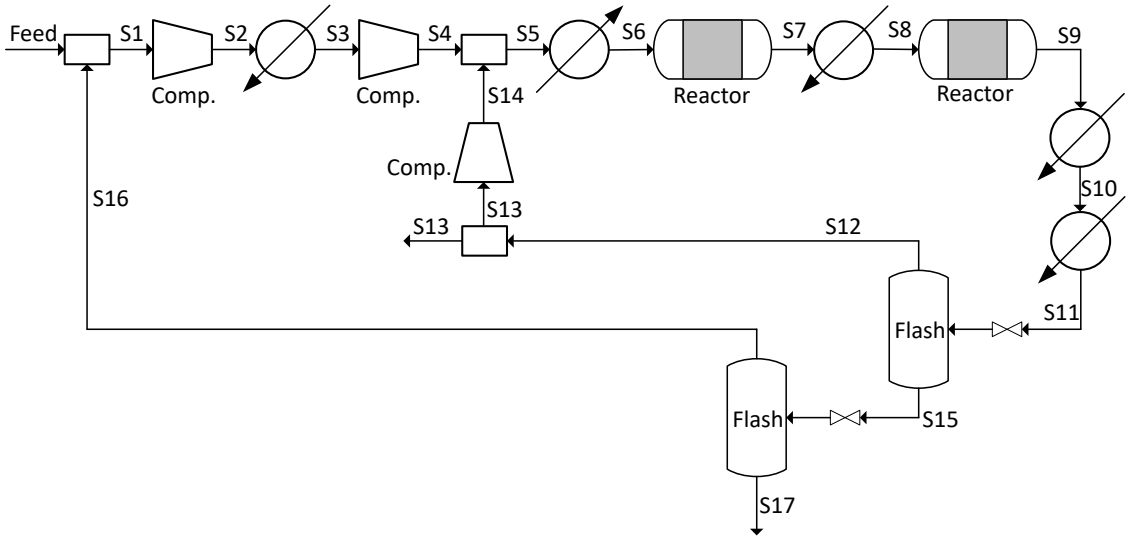

Figure 10.1: Diagram of the ammonia synthesis process [29].

The effluent of the first reactor, S7, is cooled and sent to a second reactor, in order that the equilibrium be shifted towards ammonia production [83]. The stream exiting the second reactor (S9) is coooled again and flashed to separate the ammonia (S15) from the unconverted hydrogen and nitrogen (S12), which are purged, recycled, and recompressed.

The ammonia-rich stream (S15) is flashed a second time at low pressure to further purify the product (S17). The low-pressure recycled reactants (S16) are combined with the feed stream and undergo two-stage compression with intercooling before being combined with the vapor from the first flash (S14). The combined stream (S5) is then heated to the reaction temperature and fed to the reactor sequence. The gas-phase packed-bed catalytic reactors are modeled as plug-flow reactors, following the methodology given by Elnashaie et al.[83]; variations in the radial direction are assumed to be negligible. The Temkin rate expression [77, 83] is used for the ammonia production reaction rate:

$$R_{NH_3} = k_4\left[K_a^2\phi_{N_2}(\phi_{H_2}^3/\phi_{NH_3}^2)^{\alpha} - (\phi_{NH_3}^2/\phi_{H_2}^3)^{1-\alpha}\right] \quad (10.2)$$

$$k_4 = k_{4,0}\exp(-E_4/RT) \quad (10.3)$$

$$\log(K_a) = -2.69\log(T) - 5.52\times10^{-5}T + 1.85\times10^{-7}T^2 + 2001.6T^{-1} + 2.69 \quad (10.4)$$

where R_{NH_3} is the reaction rate in kmol of NH_3 per hour cubic meter of catalyst bed. The nominal value of the pre-exponential factor $k_{4,0}$ is dependent on the catalyst and is assumed here to be 8.85×10^{14} kmol/(m^3h). The fugacities ϕ of the components are calculated using the accepted correlations given by Elnashaie et al. [84] The reactor is assumed to be homogeneous and operating at steady-state. The effectiveness factor for the catalyst is calculated with the empirical correlation by Dyson and Simon [77], given at 30 MPa:

$$\eta = -4.676+0.024T+4.687\chi-3.463\times10^{-5}T^2-11.280\chi^2+1.541\times10^{-8}T^3+10.466\chi^3 \quad (10.5)$$

where η is the effectiveness factor and χ is the conversion of nitrogen. The heat of reaction is calculated with the empirical correlation by Strelzoff [243]:

$$\Delta H_r = (-0.55+\frac{836.61}{T}+\frac{459.73\times10^6}{T^3})P - 5.35T - 2.5\times10^{-4}T^2 + 1.07\times10^{-6}T^3 - 9157.09 \quad (10.6)$$

where ΔH_r is the heat of reaction in calories per mol. The pressure in the reactor is assumed to vary only along the axial dimension, and the pressure drop is calculated using the Ergun equation [85] for one-dimensional flow, given in (4.39) . The one-dimensional mass and energy balances can then be written for the reactor model:

$$\frac{\partial \chi}{\partial V} = \frac{\eta R_{NH_3}}{2F^0_{N_2}} \tag{10.7}$$

$$\dot{m} C_p \frac{\partial T}{\partial V} = -\eta R_{NH_3} \Delta H_R \tag{10.8}$$

10.1.1 Nominal Case Optimization

The base case of the process is optimized using the result of optimization with a fixed conversion [190] as an initial guess. The feed stream consists of 110.0 mol/s H_2, 35.53 mol/s N_2, 0.89 mol/s Ar, and 1.63 mol/s CH_4 at 299.82 K and 1.013 MPa. The pseudo-transient approach (Chapter 3) was used to model and optimize the flowsheet, the latter based on an objective function maximizing profit: the revenue generated from ammonia sales less the cost of utilities (heat, compression, refrigeration, cooling water) and the ammortized cost of the reactor. All simulations in this chapter were carried out in gPROMS 4.2.1 [204] on a computer equipped with an Intel Core i7 processor at 3.40 GHz with 16 GB of RAM and the sequential quadratic programming (NLPSQP) optimization solver in gPROMS.

The axial dimension of the two reactors is discretized using a backwards finite-difference approximation. Constraints are included to ensure the product stream contains at least 99% ammonia and the purge stream does not exceed 3.4 mol/s ammonia loss. The optimal point is presented in Table 10.1 and was found in 57.3s, requiring a peak memory usage of 140.6 MB. This optimal point was used as the initial guess for all optimization under uncertainty problems in this case study.

The optimized reactor conversions are shown in Figure 10.2. The length of the both

Table 10.1: Optimized ammonia synthesis process.

Variable	Lower	Optimal	Upper
Pressure of S2 (MPa)	4.137	4.137	6.895
Pressure of S4 (MPa)	19.9	29.9	29.9
Reactor Area (m^2)	0.5	3	3
Reactor 1 Length (m)	0.1	0.98	10
Reactor 2 Length (m)	0.1	1.27	10
Temperature of S6 (K)	500	511.6	800
Temperature of S8 (K)	500	638.9	800
Temperature of S10 (K)	295.9	310.9	310.9
Purge Ratio	0.05	0.072	0.12
Reactor Sequence Conversion (%)	-	49.7%	-
Scaled Objective Function (AU)	-	3.17	-

reactors is approximately such that the ammonia synthesis reaction (10.1) reaches equilibrium at the exit of each reactor. Excess reactor length after equilibrium is reached would only affect the objective function through the ammortized capital cost and is expected to usually be minimal at the optimal operating conditions for any design.

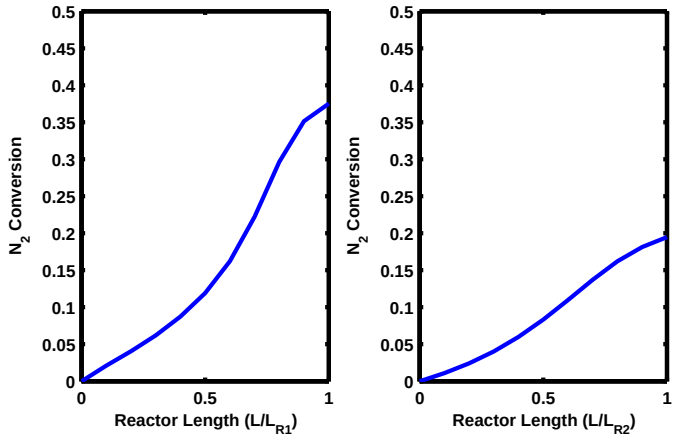

Figure 10.2: Reactor conversions in the optimized ammonia synthesis process design.

10.1.2 Multi-Scenario Optimization Under Uncertainty

The pre-exponential term $k_{4,0}$ in the Arrhenius reaction rate expression depends on the catalyst and its value can be subject to some error due to inaccurate catalyst characterization,

imperfect catalyst, or reduction in catalyst effectiveness over time. To model this, I define a variable δ such that

$$k_{4,0} = (1 + \delta)k_{4,0}^{nominal} \tag{10.9}$$

where $k_{4,0}$ is again equal to its nominal value when $\delta = 0$. For various values of δ the control variables (i.e., the process operating conditions, such as purge fraction, temperatures) can be easily changed, but optimization under uncertainty seeks the set of (constant) optimal design variables (i.e. reactor and compressor sizes). For the purpose of this case study, compressor outlet pressures are assumed to be design, rather than control, variables in order to make sure that the compressor size is the same for all realizations of the uncertain parameter.

I first consider a multi-scenario approach (8.8)–(8.10), with $n = 3$ and $[\delta_1, \delta_2, \delta_3] = [-0.2, 0.0, 0.2]$. With the scenarios weighted evenly (i.e. δ comes from a uniform distribution), the optimal point was found in 167.8s, requiring a peak memory usage of 156.6 MB. The same pseudo-transient models are used to initialize and optimize each scenario, and the constraints from the single-case optimization are included for each scenario. The decision variable values at the optimal point are presented in Table 10.2 and the values for the objective function are presented in Table 10.3.

Table 10.2: Ammonia synthesis process designed under uniformly-distributed uncertainty.

Variable	$n = 3$	$n = 5$	Dynamic
Pressure of S2 (MPa)	4.137	4.137	4.137
Pressure of S4 (MPa)	299.0	299.0	299.0
Reactor Area (m^2)	3.00	2.82	2.99
Reactor 1 Length (m)	0.98	0.99	1.01
Reactor 2 Length (m)	1.61	1.45	1.36

The reactor conversions at the optimum for the various scenarios are shown in Figure 10.3. The length of the reactors is again such that the ammonia synthesis reaction (10.1) reaches equilibrium at the exit of each reactor, suggesting that the optimal operating conditions for each scenario are again selected. The value of the objective function is almost the

Table 10.3: Multi-scenario optimized ammonia synthesis objective function. Conversion is the overall N_2 conversion in the reactor sequence.

	$n = 3$		$n = 5$	
δ	Conversion (%)	Objective (AU)	Conversion (%)	Objective (AU)
-0.2	44.7	3.07	46.5	3.06
-0.1	-	-	48.0	3.11
0.0	47.9	3.16	49.4	3.15
0.1	-	-	50.2	3.18
0.2	49.6	3.22	51.4	3.21
Expected Value		3.15		3.14

same as in the base case, where $\delta = 0.0$, suggesting that the optimal values of the design variables selected are appropriate.

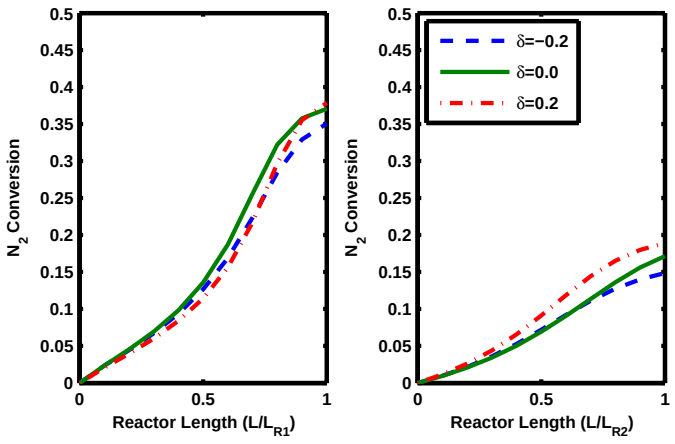

Figure 10.3: Reactor conversions in the ammonia synthesis process design optimized with three scenarios.

To examine the effects of increasing n (increasing the accuracy of the approximated expected value), I consider a multi-scenario approach (8.8)–(8.10) with $n = 5$. The values of δ are again evenly spaced, such that $[\delta_1, \delta_2, \delta_3, \delta_4, \delta_5] = [-0.2, -0.1, 0.0, 0.1, 0.2]$. The scenarios are weighted evenly, and the optimization took 541.5s, with a peak memory usage of 177.8 MB. The design variable values at the optimal point are presented in Table 10.2.

10.1.3 Dynamic Formulation Optimization Under Uncertainty

The process was then optimized for the same uncertainty using the proposed dynamic optimization framework. A linear trajectory was used for the value of the uncertain variable $\delta(\hat{t})$, with $\delta(0) = -0.2$ and $\delta(t_f) = 0.2$. Pseudo-transient models were used to smooth the algebraic process constraints, and the maximum error for all results is reported as the maximum residual (%) in any steady-state algebraic equation in (8.1). Assuming a uniform distribution for δ, the objective function throughout the entire pseudo-time horizon is evenly weighted and can be plotted against the value of δ (which varies linearly with pseudo-time $\hat{t}$) without further transformation as shown in Figure 10.4. The calculations required 4606.1s of CPU time and a peak memory usage of 144.1 MB. As hypothesized by Gutierrez [105], the memory requirements in the dynamic optimization approach are lower than in either scenario-based calculation. The maximum residual in any steady-state flowsheet equation throughout the pseudo-time trajectories is 0.07%.

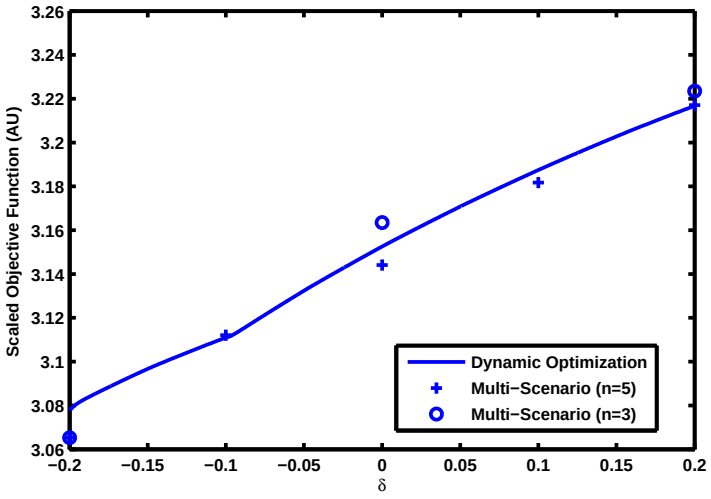

Figure 10.4: Objective function values for the ammonia synthesis process designed under uncertainty.

For this case, the optimal objective function at various values of the uncertain parameter δ calculated with all three methods are very similar. Nevertheless, the optimal values of the process design variables, shown in Table 10.2, depend largely on the discretization method used. Like the values of the objective function, the optimal values of the control variables from the dynamic optimization are given as continuous trajectories rather than as a value for each scenario, giving the optimal control variable values with which the process should be run for all values of the involved uncertain parameter(s). The same optimization problems are repeated, with δ assumed to come from a normal distribution ($\delta \sim N(0, 0.2)$) instead of a uniform distribution, and the optimal values of the design variables are shown in Table 10.4.

Table 10.4: Ammonia synthesis process designed under normally-distributed uncertainty

Variable	$n = 3$	$n = 5$	Dynamic
Pressure of S2 (MPa)	4.137	4.137	4.137
Pressure of S4 (MPa)	299.0	299.0	297.0
Reactor Area (m^2)	3.00	3.00	3.00
Reactor 1 Length (m)	1.00	1.01	1.13
Reactor 2 Length (m)	1.64	1.62	1.47

The computational times and memory requirements for the optimizations in this case study are collected in Table 10.5. The CPU times used by the optimization solver are largely dependent on the number of iterations required and do not very much between the multiple-scenario and dynamic optimization approach. The total CPU time used in each optimization was dominated by the time used by the DAE solver, which is expectedly much longer for the dynamic optimization formulation, as the pseudo-time horizon of integration is much longer than just required for initialization.

Table 10.5: Computational statistics for the ammonia synthesis process optimizations. D.O. stands for dynamic optimization

Optimization	CPU Time (Total)	CPU Time (Opti)	Max Memory Usage	Max Residual
Uniform ($n = 3$)	167.8s	0.047s	156.6 MB	-
Uniform ($n = 5$)	541.5s	0.156s	177.8 MB	-
Uniform (D.O.)	4606.1s	0.218s	144.1 MB	0.03%
Normal ($n = 3$)	489.7s	0.078s	157.3 MB	-
Normal ($n = 5$)	762.7s	0.468s	177.7 MB	-
Normal (D.O.)	17362s	0.343s	143.5 MB	0.08%

10.2 Case Study 2: Design of Dimethyl Ether Plant

In this case study, I revisit the dimethyl ether (DME) synthesis process described in Chapter 4, adapted from Appendix B of Analysis, Synthesis, and Design of Chemical Processes by Turton et al. [264]. The process is again continuous and converts methanol to dimethyl ether (DME) by the following reaction:

$$CH_3OH \rightarrow (CH_3)_2O + H_2O \qquad \text{(4.30 revisited)}$$

While Chapter 4 considered the optimization of an intensified DME process, using a quenched reactor and dividing-wall column, I now consider the conventional base case process [264]. The process flowsheet is shown in Figure 10.5.

The indirect sequence configuration has been found to be better (in terms of the objective function for this case study) than the direct sequence configuration [193], so only the indirect sequence flowsheet is considered. The process again begins by combining a methanol feed stream (S1) with the recycle stream (S9) and vaporizing the combined stream (S2) using medium pressure steam (MPS) and a feed-effluent heat exchanger. The vaporized stream (S4) is then sent to a reactor where the methanol is dehydrated to form DME. The outlet of the reactor (S5) is passed through the heat exchanger, and the stream (S6) is further chilled with cold water (if necessary) and then fed to the indirect sequence separation system, where DME product is recovered and methanol is separated and recycled (S9).

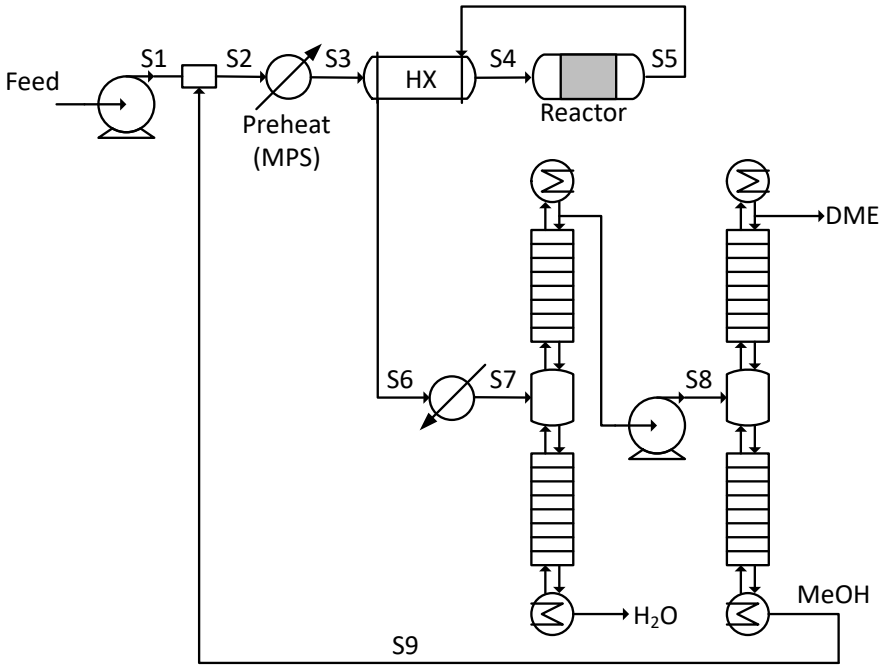

Figure 10.5: Diagram of the conventional DME process.

I model the reactor as a CSTR, using the model given in Section 4.3.1. The process has the same objective function OBJ to be minimized, defined in (4.56) and reproduced here:

$$OBJ = 63.88 \times 10^{-6}(Q_{reb} + Q_{mps}) + NT + 20 \times 10^{-6}(Q_{cond} + Q_{chill}) + 0.5V_r \quad \text{(4.56 revisited)}$$

Bypass efficiencies [71, 193], described by (4.25) and (4.28), are used to model and optimize the design of the distillation columns, i.e. select the numbers of stages and feed stage locations (given as the number of equilibrium stages from the bottom of the column). The feed flow rate is 260 mol/s and the feed stream is composed of 99 mol% methanol and 1 mol% water. Constraints are included to ensure that the feed to the columns cannot be cooled below 300 K (temperature of cooling water), the minimum temperature difference in the heat exchanger is 1 K, the DME product flow rate is at least 128 mol/s, and the product has a purity of at least 99.9 mol%. Additionally, to avoid reverse or side reactions,

the reactor temperature is again constrained to be below 390°C and the recycle stream is constrained to have a maximum of 1% DME. The UNIQUAC physical properties package was used to model the thermodynamic properties.

10.2.1 Nominal Case Optimization

The steady-state time relaxation method [190] was used to optimize the process flowsheet, and the optimal design is presented in Table 10.6. The optimal solution for the base case process was found in 204.1s of CPU time, with a maximum memory usage of 171.4 MB.

Table 10.6: Optimized conventional DME synthesis process.

Variable	Lower	Optimal	Upper
Pressure of S1 (MPa)	12.0	15.5	15.5
Temperature of S3 (K)	298.15	426.62	-
Reactor volume (m^3)	2π	54.76	54π
Reboiler 1 Duty (MW)	0	2.57	-
Reboiler 2 Duty (MW)	0	1.54	-
Reboiler 1 Pressure (MPa)	5.0	6.45	10.0
Pump ΔP	0	0	2
Column 1 Stages	0	21	30
Column 2 Stages	0	11	20
Column 1 Feed Location (from bottom)	0	9	15
Column 2 Feed Location (from bottom)	0	6	10
Reflux Ratio 1	0	1.46	10
Reflux Ratio 2	0	0.89	10

At the optimal solution, the reactor pressure and temperature reach their upper bounds (15.5 bar and 663 K respectively), giving a 69% conversion of methanol. It is also noteworthy that the heat exchanger pinch and DME in recycle stream reach their bounds of 1 K and 1 mol%, respectively. The optimization was run with a wide range of initial guesses, and the solution of the base case optimization problem was used as the initial guess for all of the ensuing design under uncertainty optimizations.

10.2.2 Multi-Scenario Optimization Under Uncertainty

The reaction rate for methanol dehydration to DME depends on the catalyst and can be subject to some error due to inaccurate catalyst characterization, imperfect catalyst, or reduction in catalyst effectiveness over time. Therefore, I define a variable δ such that

$$R_r = (1 + \delta)R_r^{nominal} \tag{10.10}$$

where $R_r^{nominal}$ is the nominal value, given by (4.31), when $\delta = 0$. For various values of δ the control variables (i.e., the process operating conditions, such as reflux ratios, reboiler duties) can be easily changed, optimization again seeks the set of optimal design variables (i.e. reactor and heat exchanger sizes). For this case study, the reactor pressure (the higher pressure in the process) is assumed to be a design, rather than control, variable in order to ensure appropriate pump sizing and vessel/piping thickness.

I first consider a multi-scenario approach (8.8)–(8.10) with $n = 3$ and $[\delta_1, \delta_2, \delta_3] = [-0.1, 0.0, 0.1]$. With the scenarios weighted evenly (i.e. δ drawn from a uniform distribution), the optimal point was found in 716.2s, requiring a peak memory usage of 265.9 MB. The same pseudo-transient models are used to initialize and optimize each scenario, and the constraints from the single-case optimization are included for each scenario. In addition, a constraint is included to ensure that the calculated heat exchange area required for the heat exchanger is the same in all scenarios.

The multi-scenario optimization provides some interesting insights: the DME concentration of the recycle stream does not reach its bound of 1 mol% in the case where the reaction rate is decreased, as less DME is produced. Although the reaction proceeds faster in the case of $\delta = 0.1$, the conversion of methanol is only slightly improved (70% conversion) because the process is limited by how much DME can be returned in the recycle stream in order to avoid side reactions. Consequently, the optimized expected value of the objective

function (using the multi-scenario approximation) is worse than in the base case (it is larger by 10.71). It is also noteworthy that the design of the columns, i.e. the number of stages and the feed locations, is unchanged from the optimal base case solution.

I expect the approximated optimal expected values of the design variables and the objective function converge to their actual expected values as the number of scenarios increases [12], and I examine the effects of increasing the number of scenarios n with $\delta_1, ..., \delta_n$ evenly spaced between -0.1 and 0.1 (equal width discretization [86]). The computational times for the optimizations are shown in Figure 10.6 and the peak memory usages for the optimizations are shown in Figure 10.7. The multi-scenario optimization problems also repeated with the uncertain variable δ drawn from a normal distribution, δ $N(\mu = 0, \sigma = 0.05)$. The parameter uncertainty space was still discretized with an equal width discretization between -0.1 and 0.1, encompassing 95% of the CDF of the probability distribution of δ.

10.2.3 Dynamic Formulation Optimization Under Uncertainty

The process was then optimized for the same parameter uncertainty using the proposed dynamic optimization framework. A linear trajectory was used for the value of the uncertain variable $\delta(\hat{t})$, with $\delta(0) = -0.1$ and $\delta(t_1) = 0.1$. Pseudo-transient models were used to initialize the process model at each optimization iteration and to smooth the algebraic process constraints, and the maximum error for all results is reported as the maximum residual (%) in any steady-state algebraic equation. For a uniformly distributed uncertain δ, the calculations required 21549s of CPU time and a peak memory usage of 185.6 MB. As expected [105], the memory requirements in the dynamic optimization approach are lower than in the scenario-based calculations (optimization with just 3 scenarios had a 265.9 MB peak memory usage, and 15 scenarios required over 1 GB). The maximum residual in any steady-state flowsheet equation throughout the pseudo-time trajectories was 0.02%. For a

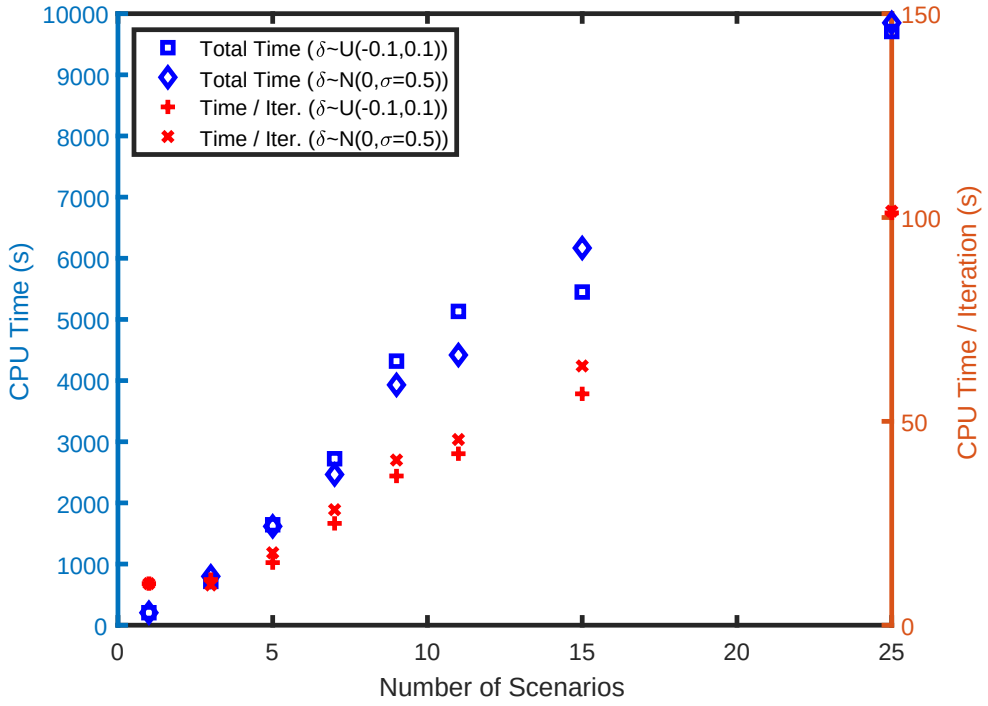

Figure 10.6: CPU times for solution of the optimization problems in the DME case study.

normally distributed uncertain δ ($\sigma = 0.05$), the calculations required 9347s of CPU time and a peak memory usage of 185.5 MB.

I found that while the dynamic optimization computations required more CPU time than the multi-scenario optimizations, the bulk of the time per iteration was spent initializing the model at the initial condition of the uncertain parameter ($\delta(0) = -0.1$) using a pseudo-transient time integration to the steady-state solution. Once the system was initialized, integration of the model (with sensitivities) through the trajectory of the uncertain parameter required relatively little time. To estimate the time in each iteration spent initializing the model, I simulated a few iterations of the optimization and found that in all cases, 93% of the CPU time was spent initializing the system, and only 7% of the CPU time was spent integrating the system following the trajectory of the uncertain parameter.

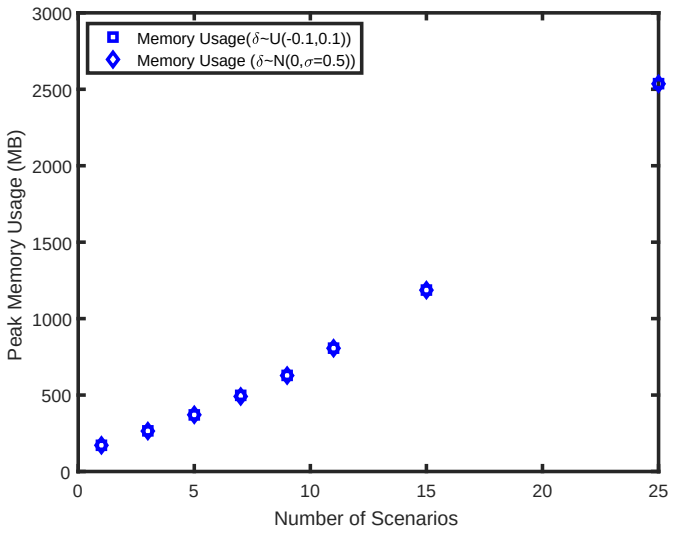

Figure 10.7: Memory requirements for solving the optimization problems in the DME case study.

While the algebraic, multi-scenario optimization problems could be solved using the time relaxation-based algorithm, allowing each iteration to be initialized from the result of the previous iteration (saving considerable amounts of time), dynamic optimization in gPROMS does not allow for this initialization strategy. Therefore, the solution time of the dynamic optimization formulation is extended by the length of time required to initialize the model at each iteration, which I believe should be closer to the length of time required to initialize each iteration of the base case optimization (since the models are approximately the same size). By replacing the amount of time spent in each iteration on initialization (93% of CPU time) with the average time per iteration for the single case optimization (10.2s), I estimate that the dynamic optimization CPU times could be reduced to 12.5% and 12.3% of their reported values for the uniform distribution and normal distribution cases respectively.

Using this estimate, the solution of the dynamic optimization formulated problems could be reduced to 2722s and 1139s for the uniform distribution and normal distribution

cases respectively (23.3s and 22.8s per iteration), which is comparable with the solution times for the multi-scenario formulation with 7 scenarios. The expected values of the objective function for both dynamic optimization cases are shown in Figure 10.8, and it can be seen that the optimized objective functions of the multi-scenario formulations converge to those of the dynamic optimization formulations as the number of scenarios increases. It is noteworthy that the dynamic optimization formulations reach approximately the same objective function value as a multi-scenario optimization with a large number of scenarios, while the peak memory usage is greatly reduced.

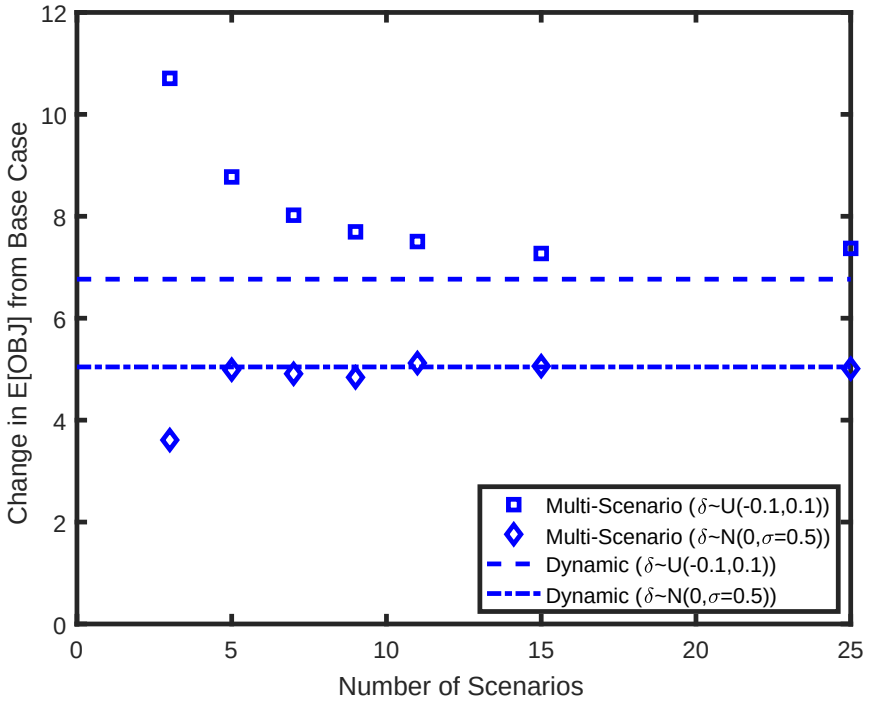

Figure 10.8: Objective function values for the DME process designed under uncertainty.

The dynamic optimization approach uses about 30% less memory than the multi-scenario approach with just 3 scenarios considered, and compared to the multi-scenario optimization with 25 scenarios, the dynamic optimization approach uses over 90% less memory.

This memory reduction would allow for more uncertain parameters to be considered; discretizing just 2 uncertain parameters with 5 discretization points (which may not be enough to accurately estimate expected values—see Figure 10.8) results in 25 scenarios, requiring over 2500 MB of memory. It is also interesting to note that the optimal design of the distillation columns provided in the solution of the dynamic optimization formulation differs from the optimal base case result, and the optimal values found using the dynamic optimization are shown in Table 10.7.

Table 10.7: Optimal DME process design variable values for reaction rate uncertainty.

Variable	Lower	Optimal ($\delta \sim U(-0.1, 0.1)$)	Optimal ($\delta \sim N(0, \sigma = 0.05)$)	Upper
Pressure of S1 (MPa)	12.0	15.5	15.5	15.5
Reactor volume (m^3)	2π	55.85	56.30	54π
Column 1 Stages	0	21	21	30
Column 2 Stages	0	11	12	20
Column 1 Feed Location*	0	9	9	15
Column 2 Feed Location*	0	6	6	10

*Feed locations given as # of trays from bottom.

10.3 Case Study 3: Design of Dividing-Wall Distillation Column

In this case study, I examine the design of a diving-wall distillation column (DWC), as in the intensified version of the DME synthesis process presented in Chapter 4. Recall that, in addition to the standard design considerations of a distillation column, a DWC features three additional degrees of freedom: (i) the liquid split across the top of the dividing wall, (ii) the side draw flow rate, and (iii) the vapor split fraction across the bottom of the dividing wall. Following Section 4.2.2, the dividing wall column can be represented as the flowsheet shown in Figure 10.9. The three additional degrees of freedom are labeled in blue.

Benzene, toluene, and p-xylene (BTX) systems are a common target for ternary separation [69]. I consider a BTX dividing-wall distillation column system with a 500 kmol/h

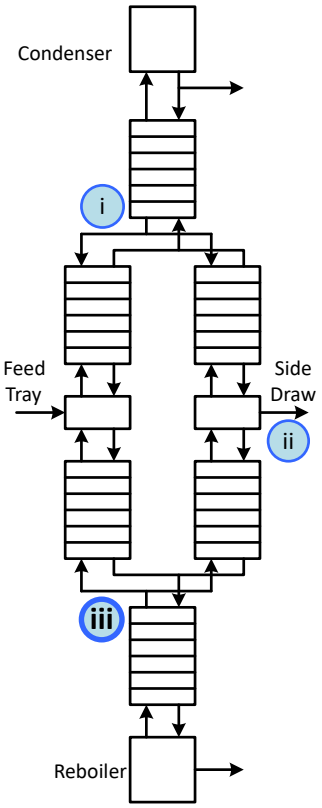

Figure 10.9: Diagram of a dividing-wall column.

feed composed of 30 mol% benzene, 40 mol% toluene, and 30 mol% p-xylene. The temperature of the column feed is 111°C, the reboiler pressure is 1.2 bar, and there is a 0.2 bar linear pressure drop through the column. The objective function for optimization is to minimize the sum of OPEX and annualized CAPEX, or the estimated annual cost of the DWC. The capital cost is annualized over a three-year period.

Bypass efficiencies are again used to optimize the number of equilibrium stages in each stage sequence, and a constraint is enforced to ensure the numbers of stages on the two sides of the dividing wall are equal. Constraints are also included to require that the distillate is at least 99 mol% benzene, the side draw is at least 99 mol% toluene, and the bottoms product is at least 99 mol% p-xylene. The pseudo-transient models for distillation

described in Section 4.2.2 are used to model the entire dividing-wall column flowsheet. The Soave-Redlich-Kwong equation of state is used to compute physical properties, with activity coefficients computed using the UNIQUAC physical properties package.

10.3.1 Nominal Case Optimization

The same steady-state time relaxation method was used to optimize the process flowsheet, and the optimal design is presented in Figure 10.10. The optimal point was found in 3189s of CPU time, with a maximum memory usage of 421.7 MB.

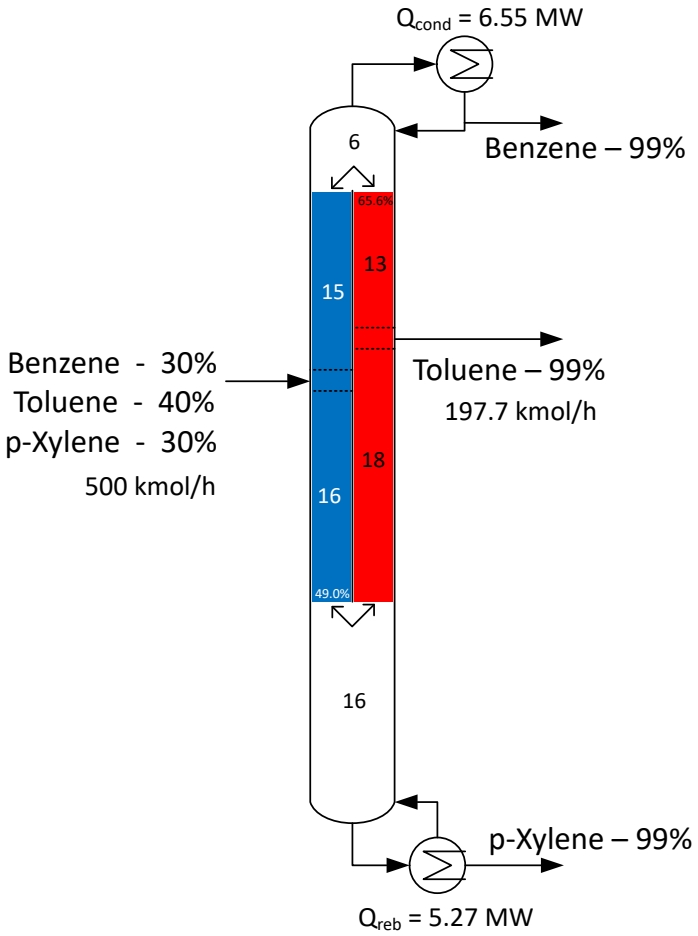

Figure 10.10: Optimized dividing-wall column design.

At the optimal point, all three outlet streams exactly reach the requisite 99 mol% purity of the corresponding component. The liquid split at the top of the dividing wall, the side draw fraction, and the bottom vapor split were treated as decision variables during column design optimization. In particular, the vapor split fraction ξ_b has an optimal value of 49.0%, where ξ_b is defined as the fraction of the vapor from the bottom stage sequence that enters the feed side of the dividing-wall section. The optimal design has an estimated annualized CAPEX of 0.42 $Million and an OPEX of 1.01 $Million, giving a total annual cost of 1.43 $Million. The compositions and temperatures throughout the column at its optimized design point are shown in Figure 10.11.

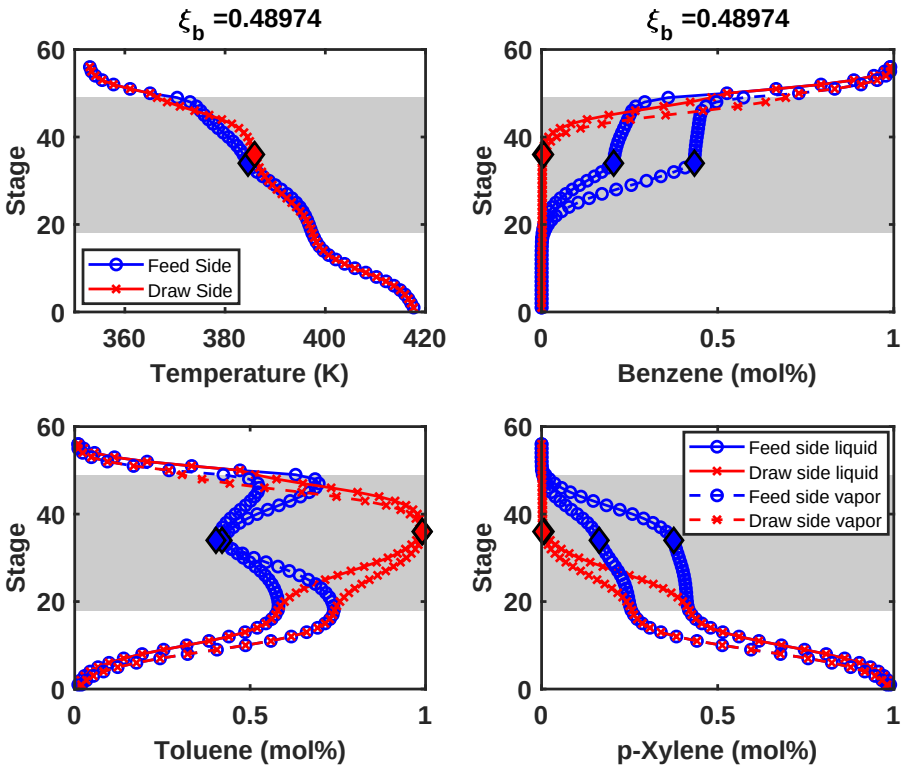

Figure 10.11: Variable profiles in optimized DWC at nominal conditions. The gray shaded area indicates the location of the dividing wall. The locations of the feed stream and side draw are marked with diamonds.

10.3.2 Operation with Variable Vapor Split Fraction

In the base-case optimization, as well as in previous studies [193, 195], the vapor split fraction (see (iii) in Figure 10.9), is determined by the optimization procedure as part of the process design. In reality, the vapor split fraction at the bottom of the dividing wall cannot be set independently or controlled by a valve. Rather, its value depends on the pressure drops on either side of the dividing wall, which are dependent on the packing structure, liquid flow rates, number of stages, etc. It is thus subject to considerable uncertainty at the design stage. To investigate the effect of this uncertainty, I define ξ_b as an uncertain variable, which I assume to be bounded to [39.0%, 59%], or a $\pm$10% absolute range around its nominally designed value. The dynamic formulation was used to prescribe a linear trajectory for $\xi_b(\hat{t})$. The pseudo-transient models were used to smooth the algebraic possible constraints.

Simulation of this trajectory from $\xi_b(\hat{t} = 0) = 39.0\%$ to $\xi_b(\hat{t} = t_f) = 59.0\%$ showed that the value of the vapor split fraction has a significant effect on column operation. Figure 10.12 shows the product stream purities found during this simulation.

Furthermore, Figure 10.13 shows profiles within the DWC when $\xi_b = 59.0\%$, constituting an extreme deviation from the nominal design. These profiles show that changes in the vapor split fraction affect the performance of the entire column. The designed side draw location is no longer at the stage with highest toluene purity, and the p-xylene purity at the bottom of the column is greatly decreased compared to Figure 10.11.

Deviations of ξ_b away from its designed value lead to significant decreases in the purities of all three product streams (Figure 10.12). Consequently, recourse variables ($\mathbf{u}$) must be adjusted to meet the product purity constraints. To adjust process operations, it is natural to control the distillate purity by manipulating the reflux ratio (or equivalently, the condenser duty). Similarly, the bottoms product purity can be controlled by manipulating

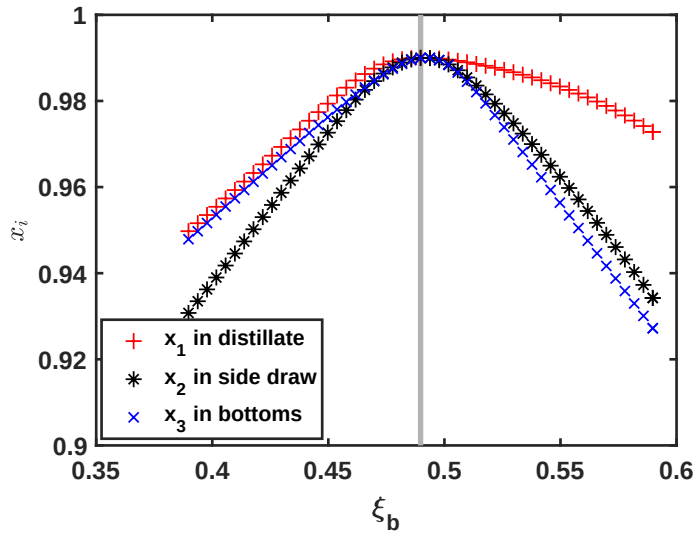

Figure 10.12: DWC product purities for variable vapor split fraction. The gray line denotes the nominal, optimal operating point.

the boil-up rate/reboiler duty. Two degrees of freedom remain to control the side draw purity: the side draw fraction and the liquid split at the top of the dividing wall. A dynamic optimization problem of the form in (9.5)–(9.8) was solved to obtain the optimal profiles of these two degrees of freedom as a function of $\hat{t}$, or equivalently, ξ_b.

With the above control strategy, the purities of benzene, toluene, and p-xylene respectively in the distillate, side draw, and bottoms product are maintained at 99 mol% for all values of ξ_b. However, the operating cost for the column increases significantly. The contributors to operating cost (condenser and reboiler duties) are shown in Figure 10.14 as a function of ξ_b. As expected, the cost reaches a minimum near the nominal design point of $\xi_b = 49.0\%$ and increases with deviations away from this point.

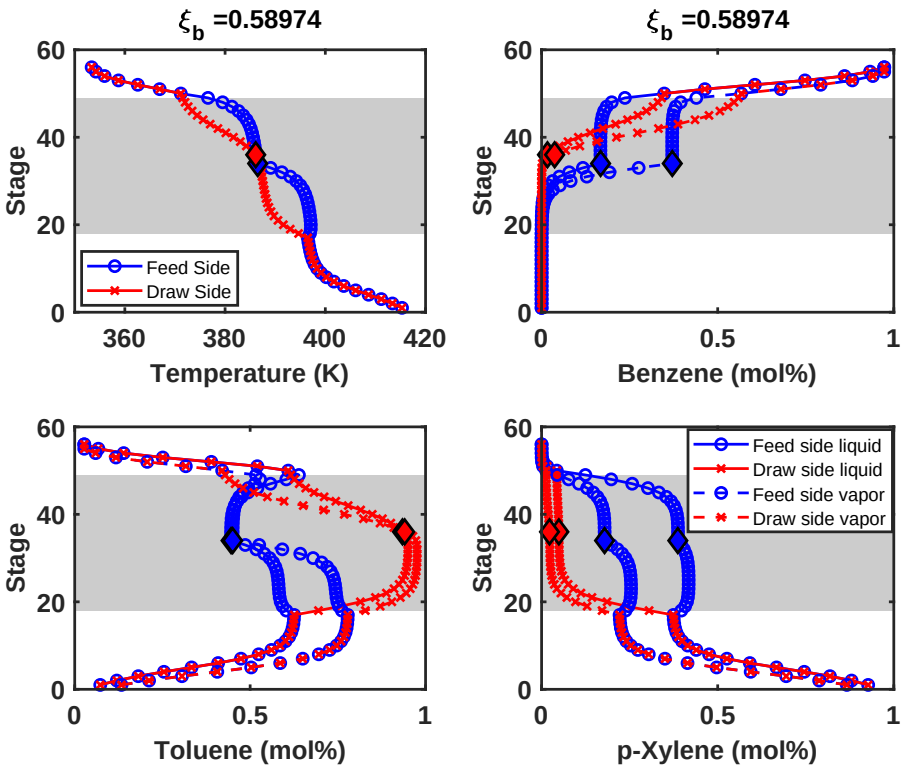

Figure 10.13: Variable profiles in optimized DWC with increased ξ_b. The gray shaded area indicates the location of the dividing wall. The locations of the feed stream and side draw are marked with diamonds.

Assuming the true value of ξ_b is uniformly distributed, or $\xi_b \sim U(39.0\%, 59.0\%)$, the expected value of the operating cost is 1.08 \$Million, giving a total annual cost of 1.51 \$Million for the DWC. The CAPEX does not change, as only the control strategy was changed in order to meet product purity requirements.

10.3.3 Design for Variable Vapor Split Fraction

While the previous section investigated how an operating strategy using $\mathbf{u}$ as a function of ξ_b could be used to meet process constraints, I expect that further economic benefits could be derived by optimizing the process design ($\mathbf{d}$) considering its variable operation.

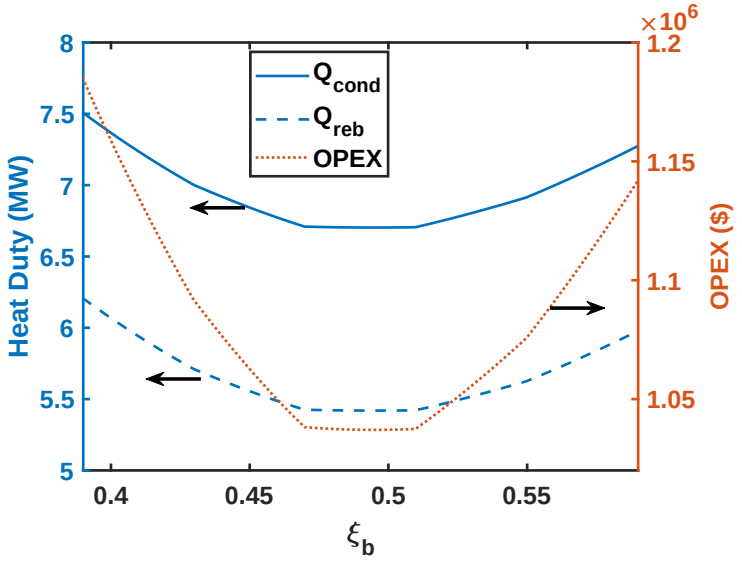

Figure 10.14: Heat duties and OPEX to control product purities for variable vapor split fraction.

The process design and operational strategy were then optimized simultaneously using the same dynamic optimization formulation (9.5)–(9.8). Because the value of ξ_b is itself a design variable in $\mathbf{d}$ for process design, I assumed that ξ_b is uniformly distributed within $\pm 10\%$ of its optimal value. This formulation effectively allows the vapor split fraction to be designed (e.g., via packing structures), but still ensures that the DWC is robust to operation in cases where k_b deviates. The optimal design has $\xi_b = 55.3\%$ and is shown in Figure 10.15.

The column has one extra tray, and the locations of the feed and side draw are altered. Moreover, the reboiler and condenser duties are higher at the nominal operating point of ξ_b. The optimal design has an estimated annualized CAPEX of 0.44 \$Million and an OPEX of 1.01 \$Million. The total cost is 1.46 \$Million, compared to the 1.43 \$Million cost estimate for the column designed without considering uncertainty. However, the new column design is robust to changes in ξ_b, and the expected value of the OPEX for $\xi_b \sim U(45.3\%, 65.3\%)$ only increases to 1.02 \$Million. Figure 10.16 compares the robustness of the two designs to

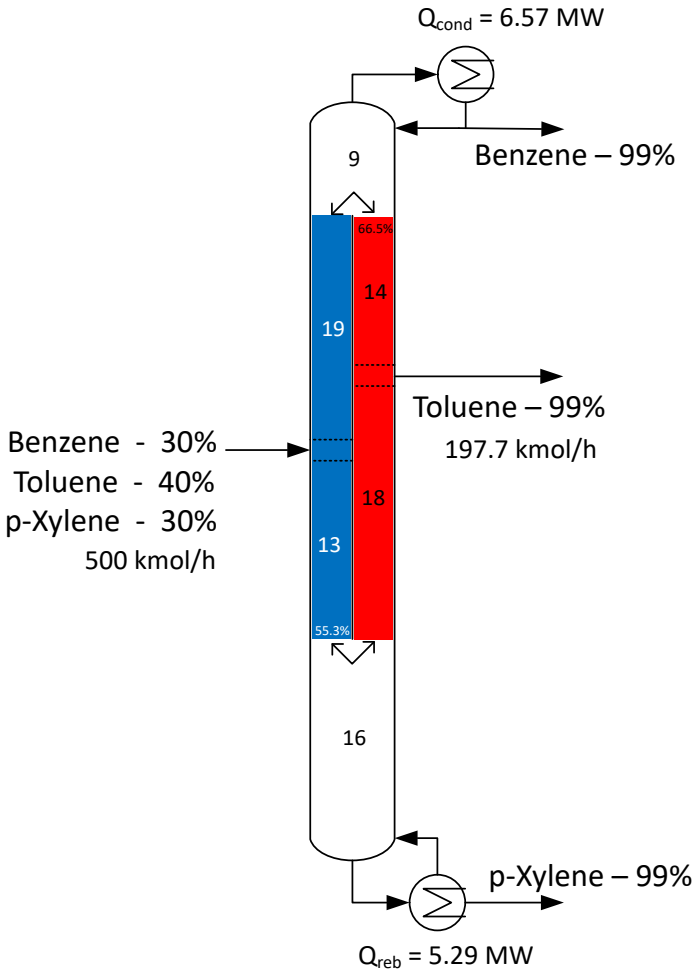

Figure 10.15: Optimized dividing-wall column design for vapor split uncertainty.

deviations in ξ_b in terms of OPEX.

As a result of the lower impact of vapor-split deviations in OPEX, the expected annual cost of the DWC designed uncertainty is 1.46 \$Million. This constitutes a 3.3% annual cost savings compared to the expected annual cost of 1.51 \$Million for the DWC designed without considering uncertainty (controlling for product purities of 99 mol%). The profiles of the temperatures and compositions in the DWC at its nominal conditions and with a 10% increase in ξ_b are again shown in Figures 10.17 and 10.18. The similarity between the

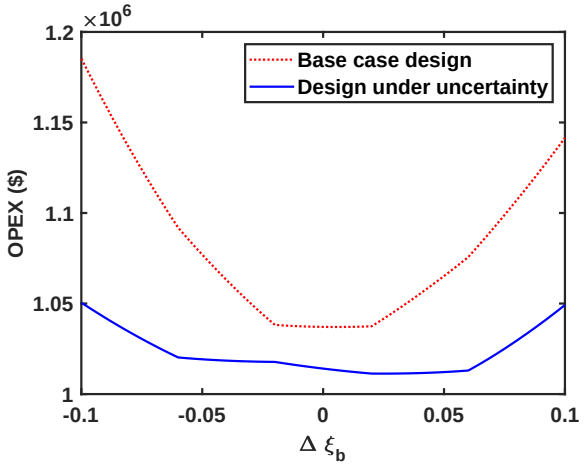

Figure 10.16: OPEX for variable vapor split fraction in designs found with and without considering uncertainty.

profiles in the figures confirms that the proposed control scheme and strategy for optimization and uncertainty succeed in a DWC that is robust to this case of parametric uncertainty.

10.4 Summary

This chapter examined the optimal design of several prototypical processes following the reaction/separation/recycle paradigm for practical examples of uncertainty. I showed how this important class of problems can be reformulated as a dynamic optimization problem using the strategies described in Chapter 9, and using the pseudo-transient strategies in Chapter 3 to smooth the algebraic path constraints. I studied the design of an ammonia production plant and revisited the dimethyl ether production process, investigating their optimal designs for uncertainty in reaction rate (e.g., due to variable catalyst performance). Furthermore, I investigated the optimal design of dividing-wall distillation columns for robustness to the vapor split fraction at the bottom of the wall, an important open question for this intensified process unit.

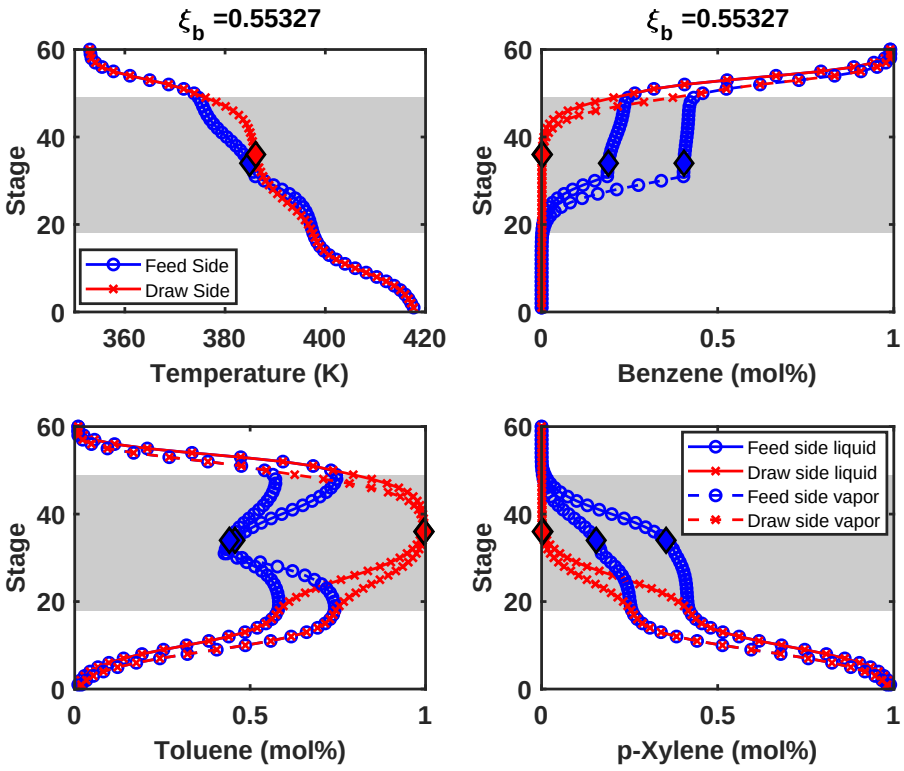

Figure 10.17: Variable profiles in DWC optimized under uncertainty at nominal conditions. The gray shaded area indicates the location of the dividing wall. The locations of the feed stream and side draw are marked with diamonds.

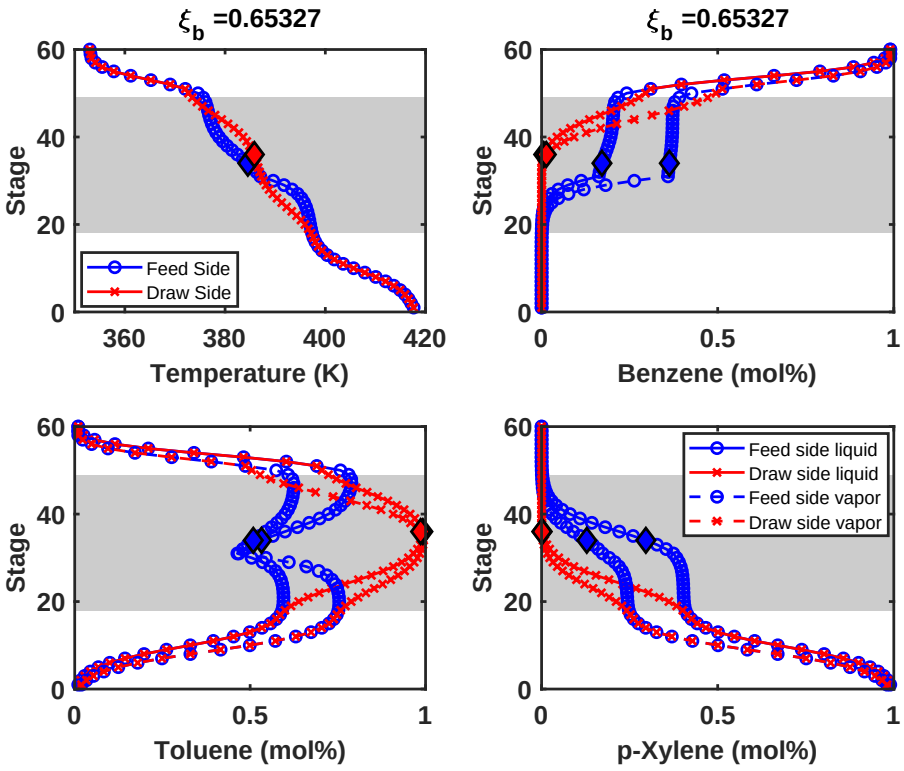

Figure 10.18: Variable profiles in DWC optimized under uncertainty with increased ξ_b. The gray shaded area indicates the location of the dividing wall. The locations of the feed stream and side draw are marked with diamonds.

Chapter 11

Modeling and Optimization of the PRICO Natural Gas Liquefaction Process for Variable Feed[†]

In this chapter, I revisit the PRICO natural gas liquefaction process introduced in Chapter 5. While the composition of the natural gas feedstock was assumed fixed, in reality natural gas feed compositions can be uncertain and/or change in time, depending on the reservoir, and liquefaction systems may not always operate at nominal design conditions. Consideration of variations in feed composition is especially important in the natural gas supply chain, where products are typically sent to customers with little or no processing [152]. With this motivation, I investigate the design of the PRICO natural gas liquefaction process under feed composition uncertainty, modeled by treating the concentration of the natural gas to be liquefied as a parametric uncertainty. This chapter first introduces a pseudo-transient compressor model based on compressor dynamics/curves to account for variable efficiency and instability (i.e., operation past the surge line) at different operating points. Then, the flowsheet including compressor and MHEX sizing is optimized for the case of feed concentration uncertainty, with both single-stage and two-stage compression configurations. The presentation in this chapter follows closely the study in Tsay and Baldea (2017) [253].

[†]The contents of this chapter are largely based on the following publication: C. Tsay and M. Baldea. Scenario-free optimal design under uncertainty of the PRICO natural gas liquefaction process. Ind. Eng. Chem. Res., 57(17):5868–5880 , 2019. C.T. is the primary author of the manuscript.

11.1 Motivation

Many efforts have been dedicated to improving the energy efficiency of liquefaction processes, including LNG flowsheet simulation and optimization advances [122, 132, 166, 192, 260] by the Process Systems Engineering community. Single mixed refrigerant (SMR) liquefaction processes, such as the PRICO process (also described in Chapter 5), have been of particular interest, due to their relatively simple designs and small footprints making them cost effective for offshore and stranded LNG plants [132]. For example, Lee et al. [149] found that the composition of the refrigerant is the most significant variable for SMR process design. Aspelund et al. [8] used the Nelder-Mead algorithm for local searches in a simulation-based optimization scheme. Tak et al. [245] compared the optimized designs for various compression configurations. Khan and Lee [132] implemented a particle-swarm optimization paradigm. Jensen [118] considered a nonlinear objective function to balance capital and operating cost trade-offs.

In most of these studies, the specific work of the liquefaction cycle (i.e., the amount of energy required per unit mass of LNG produced) plays a large role in the objective function. These studies generally consider steady-state optimization for a known natural gas feed [122, 192] or a few scenarios involving different natural gas feed compositions [8, 245]. While assuming fixed natural gas composition(s) facilitates the comparison among the specific work values associated with optimal SMR process designs reported previously, in reality natural gas feed compositions can change in time, depending on the reservoir, and liquefaction systems may need to operate at variable conditions depending on feed.

Recognizing these challenges, Li et al. [152] considered the optimization of natural gas production systems at the infrastructure level, using a scenario-based stochastic programming approach and a nonconvex decomposition method to maximize the expected process

profit. The benefits that were identified for designing the natural gas infrastructure for variable feed compositions suggest that the design of LNG liquefaction processes should also consider this uncertainty. While optimization under uncertainty often results in process designs that can differ considerably from the nominal case [191, 259], feed concentration uncertainty has rarely been considered in the optimization of SMR liquefaction processes and remains an open research question.

11.2 Natural Gas Liquefaction Process Modeling

NG liquefaction processes convert natural gas into LNG using a combination of heat exchangers, compressors, flash tanks, and valves [8, 166]. They are generally categorized as (i) mixed-refrigerant processes, based on adjusting coolant composition to match the LNG cooling curve, (ii) cascade processes, employing multiple refrigeration cycles to limit the mean temperature difference, and (iii) mixed-fluid cascade processes, combining both techniques to further improve energy efficiency. In the first category, Figure 11.1 shows a cascaded LEC (Liquefied Energy Chain) process flowsheet [207, 280]. As illustrated by this example, compressors and MHEXs are typically the central units for LNG processing plants, and their careful, optimization-oriented design is crucial to maximize process energy efficiency (minimize specific work).

11.2.1 Pseudo-Transient Multistream Heat Exchanger Model

Multistream heat exchangers (MHEXs), which improve process efficiency by allowing multiple hot and cold streams to exchange heat in a single device, introduce several challenges into LNG process modeling: (i) the minimum temperature approach constraints between any two streams cannot be written explicitly in terms of the inlet and outlet stream temperatures; (ii) the temperature driving force along MHEXs is often very small; and (iii)

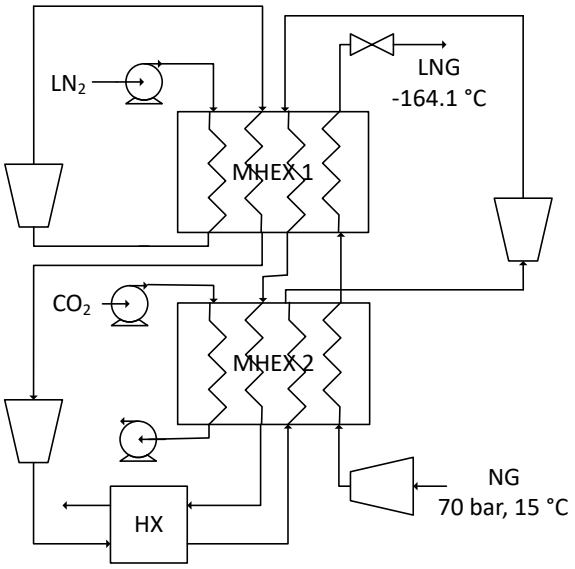

Figure 11.1: Process flow diagrams for a LEC liquefaction process [280]. The LEC process employs CO_2 from industrial sources and N_2 from a connected air separation plant as refrigerants. The outlet CO_2 stream is typically used for enhanced oil recovery; N_2 is released to the atmosphere.

the physical properties of the streams (notably, heat capacity) cannot be assumed to be constant over the typically wide temperature range of the streams present in the exchanger. Nevertheless, in Chapter 5, I showed that detailed, geometrical MHEX mathematical models can be seamlessly incorporated into flowsheet simulation and optimization using a pseudo-transient continuation approach.

To address process design under uncertainty (which requires more computational effort), this chapter employs the simpler, thermodynamic-based MHEX mathematical model proposed for process optimization by Pattison and Baldea [192]. The model is not specific to spiral-wound MHEXs (as in Chapter 5), and it relies on discretizing the exchanger into a finite number of enthalpy segments. As is often true in practical application, the ordering of MHEX inlet and outlet streams in terms of temperature magnitude is again assumed to be invariant and known prior to simulation and optimization. The cold stream inlet and outlet

temperatures comprise the cold temperature sequence, and those of the hot streams comprise the hot temperature sequence. Once the number of points (where each point corresponds to an individual inlet or outlet stream temperature) in the cold temperature sequence N_C and the number of points in the hot temperature sequence N_H are known, the number of heat exchange intervals N_{HX} is given by

$$N_{HX} = N_C + N_H - 3 \tag{11.1}$$

Each heat exchange interval contains a fixed set of cold and hot streams, separated from its adjacent intervals by a stream inlet or outlet. The continuity of stream properties between consecutive intervals is ensured by additional equalities (11.3)–(11.4), written for each heat exchange interval (enthalpy interval) comprising inlet and outlet conditions for the set of streams (S_i) present in each enthalpy interval ($i \in INT$). INT denotes the set of all enthalpy intervals.

$$Q_i = \sum_{c \in \mathcal{C}_i} (H_{c,i,out} - H_{c,i,in}) = \sum_{h \in \mathcal{H}_i} (H_{h,i,in} - H_{h,i,out}) \tag{11.2}$$

$$H_{c,i,out} = H_{c,i+1,in}, \forall\ c \in (\mathcal{C}_i \cap \mathcal{C}_{i+1}) \tag{11.3}$$

$$H_{h,i,in} = H_{h,i+1,out}, \forall\ h \in (\mathcal{H}_i \cap \mathcal{H}_{i+1}) \tag{11.4}$$

Index $c \in \mathcal{C}_i$ represents the set of cold streams in enthalpy interval i, index $h \in \mathcal{H}_i$ represents the set of hot streams ($\mathcal{H}_i = S_i \setminus \mathcal{C}_i$), and Q_i represents the total enthalpy exchanged in enthalpy interval i. Each heat exchange interval (enthalpy interval) $i \in INT$ is then discretized into a finite number of enthalpy segments N_i, as shown in Figure 11.2.

Previous works [192, 278] suggest that model accuracy is dependent on the number of enthalpy segments selected for each interval, and more enthalpy segments should be used when the dependence of heat capacity on temperature and/or the likelihood of phase changes

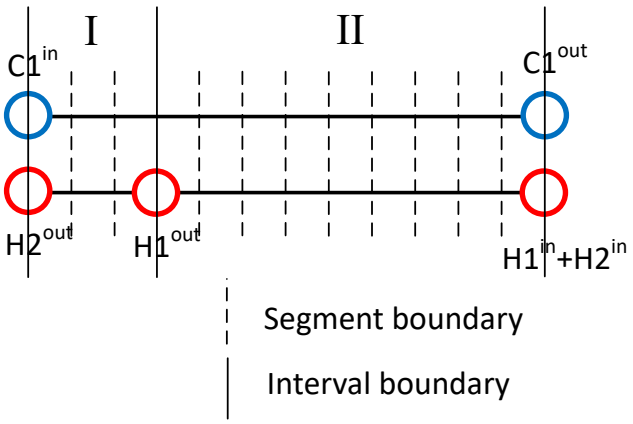

Figure 11.2: Segmentation of an example MHEX into enthalpy intervals and enthalpy segments. The MHEX features a single cold stream and two hot streams. The hot streams share a common inlet temperature, and the resultant MHEX has two associated enthalpy intervals (I & II).

are significant. The enthalpies of the cold and hot composite curves, denoted respectively as $\bar{H}_i^C$ and $\bar{H}_i^H$, can then be calculated at each segment $z_i = [0, 1, ..., N_i]$ by

$$\bar{H}_i^C(z_i) = \frac{Q_i}{N_i}(z_i) + \sum_{c \in \mathcal{C}_i} H_{c,i,in} \tag{11.5}$$

$$\bar{H}_i^H(z_i) = \frac{Q_i}{N_i}(z_i) + \sum_{h \in \mathcal{H}_i} H_{h,i,out} \tag{11.6}$$

The pressure drop for each stream ΔP_s is assumed to be known at the design stage and to vary linearly with heat duty (the pressure drop for a segment i is computed as $\Delta P_{s,i}/\Delta P_s = Q_i/\sum Q_i$). The model as presented above only provides the enthalpies of the hot and cold streams at each enthalpy segment. To calculate the composite curve temperatures at each segment, denoted respectively as $T^C(z_i)$ and $T^H(z_i)$, a set of first-order differential equations is introduced:

$$\left(\frac{H_{ref}}{T_{in}^C}\right)\tau_e \frac{dT^C(z_i)}{d\hat{t}} = \bar{H}_i^C(z_i) - \sum_{c \in \mathcal{C}_i} H^{PP}(T_c(z_i), P_c(z_i), F_c, \boldsymbol{x}_c) \tag{11.7}$$

$$\left(\frac{H_{ref}}{T_{in}^H}\right)\tau_e \frac{dT^H(z_i)}{d\hat{t}} = \bar{H}_i^H(z_i) - \sum_{h \in \mathcal{H}_i} H^{PP}(T_h(z_i), P_h(z_i), F_h, \boldsymbol{x}_h) \tag{11.8}$$

where H^{PP} is again the total enthalpy calculated by an external physical properties package. $\frac{H_{ref}}{T_{in}^{C}}$ and $\frac{H_{ref}}{T_{in}^{H}}$ are scaling factors to ensure consistent units (T_{in}^{C} and T_{in}^{H} denote the stream inlet temperatures). The model is initialized by setting the initial temperature profiles to be constant along the MHEX, with temperature values equal to the inlet conditions

$$T^{C}(z_i) = T_{in}^{C}, \forall\ z_i = [0, 1, ..., N_i] \tag{11.9}$$

$$T^{H}(z_i) = T_{in}^{H}, \forall\ z_i = [0, 1, ..., N_i] \tag{11.10}$$

While most MHEX models only account for one side of the heat exchanger [71, 122], this pseudo-transient representation provides both the hot and cold composite curves. In turn, this allows for approximating the heat exchanger area by using the required heat duty and temperature driving force between the composite curves. Assuming a heat-transfer coefficient U_i for each enthalpy interval, the area of the heat exchanger is computed as [192]:

$$A_i = \frac{Q_i}{U_i} \frac{N_i}{\sum_{z_i} T^{H}(z_i) - T^{C}(z_i)} \tag{11.11}$$

$$A_{MHEX} = \sum_{i} A_i \tag{11.12}$$

where A_i is the required area for enthalpy interval i, and A_{MHEX} is the total area of the heat exchanger. In turn, A_{MHEX} can be used to approximate the capital investment for the designed MHEX.

11.2.2 Pseudo-Transient Compressor Model

Compressors account for the majority of energy used in natural gas liquefaction, and therefore, predictive compressor models are crucial for optimization of energy consumption. Specifically in the context of optimization under uncertainty, compressor models should be able to describe changes in efficiency and stability (e.g., operation past the surge point) for a range of operating conditions. Compressors can be described by the following energy balance

equation at steady state:

$$0 = H_{in} - H_{out} + \dot{W} - Q_{ext} \tag{11.13}$$

where $\dot{W}$ is the work added to the system and Q_{ext} includes losses due to inefficiency. Iassume that the pressure at the inlet and outlet of the compressor (P_{in} and P_{out}) are design parameters, and the energy balance equation above is used to compute the outlet enthalpy (and temperature) given the inlet enthalpy. The efficiency of a compressor is defined as

$$\eta_c = \frac{H_{out}^{ise} - H_{in}}{H_{out} - H_{in}} \tag{11.14}$$

where H_{out}^{ise} is the enthalpy of the outlet stream for an isentropic process:

$$S_{in} = S(T_{in}, P_{in}, F, \boldsymbol{x}) = S(T_{out}^{ise}, P_{out}, F, \boldsymbol{x}) = S_{out}^{ise} \tag{11.15}$$

$$H_{out}^{ise} = H(T_{out}^{ise}, P_{out}, F, \boldsymbol{x}) \tag{11.16}$$

Treating the nonideality ($\eta_c < 1$) of the compressor as a sink in the energy balance (11.13), the term containing η_c is multiplied by the continuation parameter $\hat{\alpha}$ described by (3.11)–(3.3.2). A pseudo-transient compressor model is thus formulated, whereby first-order differential equations are used to calculate the unknown temperatures. The isentropic outlet temperature T_{out}^{ise} is calculated by matching the inlet and (isentropic) outlet entropies (11.15), and the actual outlet temperature T_{out} is calculated using the energy balance (11.13) and the definition of efficiency (11.14):

$$\left(\frac{S_{in}}{T_{in}}\tau_e\right)\frac{dT_{out}^{ise}}{d\hat{t}} = S_{in} - S_{out}^{ise} \tag{11.17}$$

$$\left(\frac{H_{in}}{T_{in}}\tau_e\right)\frac{dT_{out}}{d\hat{t}} = H_{in} - H_{out} + \frac{\hat{\alpha}}{\eta_c}(H_{out}^{ise} - H_{in}) \tag{11.18}$$

where T_{in} is the inlet temperature to the compressor. Most previous works regarding steady-state LNG process optimization [118, 122, 192] assumed the compressor efficiency

to be known and constant. While this assumption may be appropriate for nominal-case operation, compressor efficiency is in effect dependent on the operating regime, and variations/uncertainty in the natural gas feed composition may result in sub-optimal compressor performance.

In order to capture variations in efficiency, I now use the cubic equation given by Jensen [118] to model the characteristic curve of a compressor. Given a desired pressure ratio $R_P = P_{out}/P_{in}$, the required rotational speed N of the compressor can be computed from

$$R_P = R_{P,0} + h\left(1 + 1.5\left(\frac{\dot{m}r}{w} - 1\right) - 0.5\left(\frac{\dot{m}r}{w} - 1\right)^3\right) \tag{11.19}$$

$$h = h_0 - 1.2(h_0 + 0.5R_{P,0} - 1)(1 - Nr) \tag{11.20}$$

$$w = w_0(Nr)^{1/3} \tag{11.21}$$

$$N = \frac{Nr\sqrt{RT_{in}/MW}}{D} \tag{11.22}$$

where $R_{P,0}$, h_0 and w_0 are parameters obtained from fitting compressor characteristic curves, D is the diameter of the compressor wheel, and $\dot{m}r$ is the non-dimensionalized flow, defined as $\dot{m}r = \dot{m}\sqrt{RT_{in}/MW}D^{-2}P_{in}^{-1}$. h and w are the semiheight and semiwidth of the compressor characteristic curve, MW is the molar weight of the compressor stream, and Nr is a nondimensional speed ratio given by Saravanamuttoo et al. [219]. It is noteworthy that in many empirical expressions [118, 166], compressor rotational speed is given as a percentage (of maximum speed), rather than as an absolute velocity.

The compressor efficiency is then computed [118]

$$\eta_c = \eta_0\left(\left(1 - \left(\frac{h - h_0}{h_0}\right)^2\right) - 1000(\dot{m}r - 2w)^2\right) \tag{11.23}$$

where η_0, the nominal efficiency, is a parameter of the compressor. A set of compressor curves predicted using (11.19)–(11.22) for various rotational speeds is shown in Figure 11.3.

The surge point, where dynamic instability occurs, is located near the peak value of R_P for a given compressor rotational speed [118], plotted as black dots in Figure 11.3.

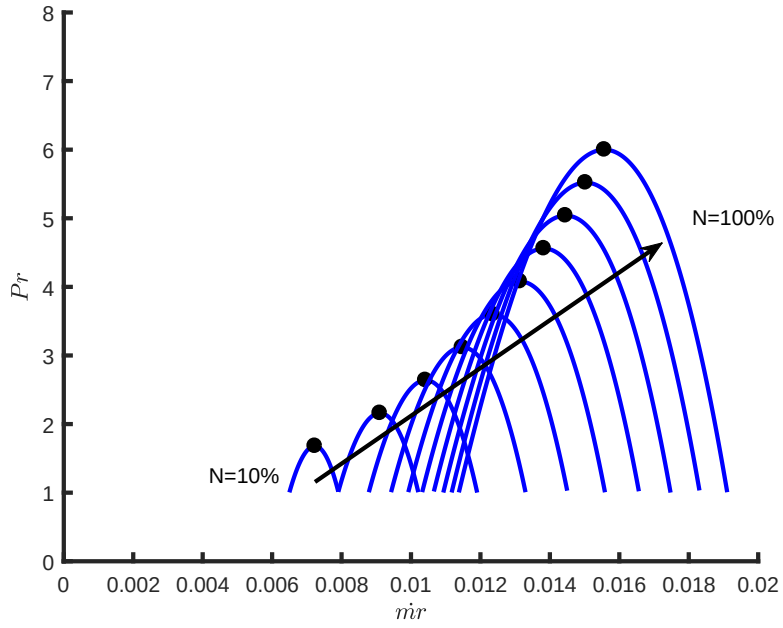

Figure 11.3: Compressor curves over a range of nondimensionalized flow for various compressor speeds. Black dots are plotted at the peak of the curve for each speed, with operation on a curve to the left of the peak representing compressor surge.

Operation to the left of this point on a curve may be possible with active (feedback) control, but generally, it is preferable to operate to the right of the surge line [99]. As proposed by Jensen [118], I define $\dot{m}r_{peak}$ as the value of $\dot{m}r$ for which the pressure ratio of the compressor characteristic curve reaches its maximum, and a dimensionless surge margin (for a given rotational speed) can be expressed:

$$\text{Surge Margin} = \dot{m}r - \dot{m}r_{peak} \tag{11.24}$$

Table 11.1: Natural gas feed composition (in mol %) under uncertainty.

	δ	Methane	Ethane	Propane	n-Butane	Nitrogen	Molar Mass
Lower Bound	-0.1	84.0	8.6	2.8	0.2	4.4	17.6 g/mol
Base Case	0	89.7	5.5	1.8	0.1	2.8	17.6 g/mol
Upper Bound	0.1	95.0	2.7	0.9	0.0	1.4	17.6 g/mol

11.3 Design of LNG Plants for Variable Feed Composition

A specific case of parametric uncertainty is now employed to capture the situation of a varying composition feed stream to a liquefaction process. For the nominal natural gas feed stream given, I consider an uncertain parameter δ drawn from a probability distribution $J(\delta)$, which represents the additional amount of methane present in the feed (in kmol/s). The flow rates of the other components are scaled accordingly, such that ethane, propane, n-butane, and nitrogen remain in the same molar proportions to each other, and the total molar mass of the natural gas stream remains constant for all values of δ. δ is assumed to be bounded between -0.1 and 0.1 kmol/s. The feed gas compositions at these bounds are shown in Table 11.1, and I note their similarity to the lean and rich gas compositions found in other works [8, 245].

A multiple-scenario optimization formulation with an equal-width discretization (δ is sampled at n equally spaced intervals between δ^L and δ^U, including the bounds) results in n realizations of δ, where the value of n is defined prior to the optimization procedure. However, I apply the dynamic optimization reformulation defined in Chapter 9 to forgo the definition of scenarios, and instead "scan" through all possible values for δ. Since the solution of the original algebraic process flowsheet model (3.1) is recovered at the steady state of a pseudo-transient model (per the static equivalence condition), I incorporate the pseudo-transient models to smooth algebraic constraints in the proposed optimization by selecting time constants such that the dynamics of the pseudo-transient process model evolve in a much faster time scale than the dynamics of the uncertain parameters.

11.3.1 Optimization Problem Formulation Details

In this work, I ignore the removal of heavy components from and pretreatment of the natural gas feed stream, including removal of water, carbon dioxide, etc. Further, I assume that the natural gas stream is passed through the full length of the MHEX, differing slightly from Chapter 5. This is motivated by the fact that the optimal solution found there involved both hot streams exiting the MHEX at nearly identical temperatures. The process flow diagram for the PRICO process is shown in Figure 11.4.

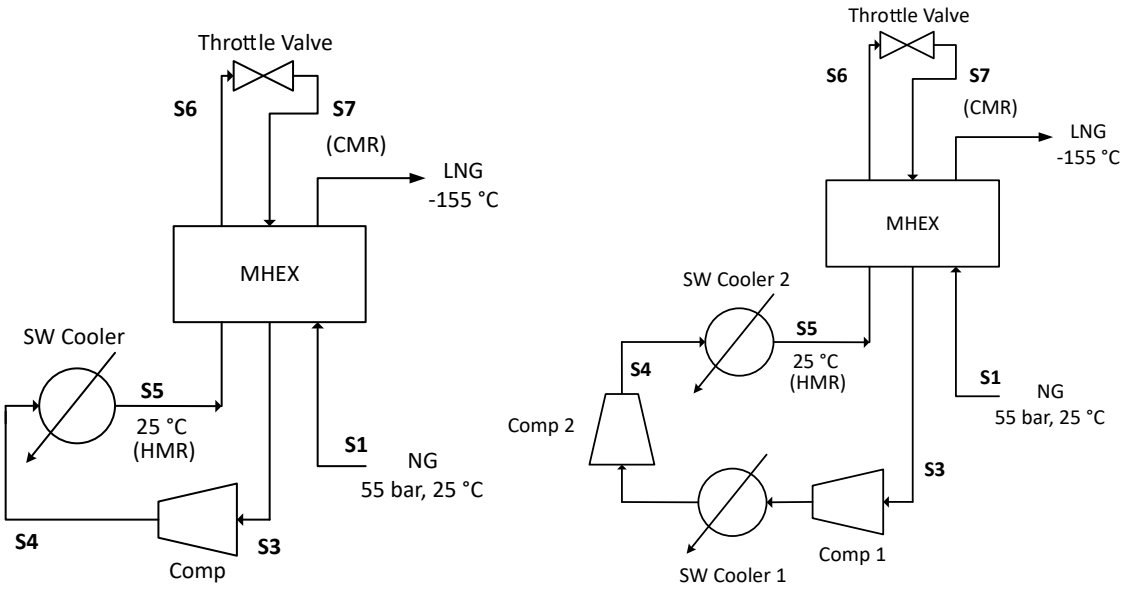

Figure 11.4: Diagrams of the simplified PRICO natural gas liquefaction process. The configuration with a single compressor is shown on the left, and the configuration with two staged compressors is on the right.

A 1 kmol/s natural gas stream (S1), nominally composed of 89.7% methane, 5.5% ethane, 1.8% propane, 0.1% butane, and 2.8% nitrogen, enters the MHEX at 25°C and 55 bar and exits the MHEX liquefied and subcooled at -155°C. The mixed refrigerant stream is composed of the same five components and is employed in a single-stage refrigeration cycle. Hot mixed refrigerant is cooled in the MHEX, expanded through a throttle valve, repassed

through the MHEX to liquefy the natural gas, and then compressed and chilled to 25°C in a salt water (SW) cooler. The pseudo-transient models described in the previous section are used to model all process units, and the Soave Redlich Kwong cubic equation of state is used to evaluate thermodynamic properties, with fluid properties calculated using mixing rules.

In addition to the challenges described in Chapter 5, natural gas feedstocks can be of various compositions, and variations in natural gas composition affect the specific work, as the specific enthalpy of each component is different from the others. As a consequence, the gas compositions considered in past studies that focused on simulation and optimization of LNG plants vary considerably (e.g., [8, 245]).

The design decision variables considered here for the PRICO liquefaction process (Figure 11.4) include compressor diameter (as described in Section 11.2), refrigerant flow rate, refrigerant composition, and the high and low pressures of the refrigeration cycle. The single degree of freedom of the MHEX is the exit temperature of the cold refrigerant stream (S6 in Figure 11.4), which must be completely vaporized to avoid damaging the compressor. With these considerations, the following constraints are included in the optimization problem:

- The temperature of the cold refrigerant exiting the MHEX (S6) must be at least 10°C greater than the dew point
- The minimum approach temperature in the MHEX must be at least 1.2°C
- The compressor does not operate past 100% of its maximum rotational speed
- The compressor surge margin, as given by (11.24), is positive

The compressor surge margin is merely constrained to be positive because operation beyond this point is in principle still possible with active control and some design modifications [99, 118]; however, the bound can easily be increased to provide a more conservative design.

The models described in Section 11.2 are used to simulate and optimize the flowsheet. For optimization of the PRICO process, two objective functions are considered: (i) minimization of the compressor work [192] and (ii) minimization of a nonlinear objective function incorporating the size of the MHEX [118, 119]. The objective functions are respectively denoted as $OBJ1$ and $OBJ2$. As will be described in the following sections, optimization solutions were computed by minimizing either $OBJ1$ or $OBJ2$.

$$OBJ1 = W_{comp} \tag{11.25}$$

$$OBJ2 = W_{comp} + C_0(A_{MHEX})^{0.65} \tag{11.26}$$

The size of the MHEX for $OBJ2$ is calculated assuming an average heat transfer coefficient of $U = 500$ $\mathrm{Wm^{-2}K^{-1}}$, and the cost factor (C_0) was selected to be 2135 $\mathrm{Wm^{-1.3}}$, as given by Jensen and Skogestad [119].

Compressor and MHEX Considerations. Flowsheet optimization was carried out for design configurations with both single- and two-staged compression (the latter with intercooling, similar to the configuration proposed by Tak et al. [245]). The process flowsheets for both configurations are shown in Figure 11.4.

The nominal efficiency of all compressors was assumed to be 82.2%, and compressor curve parameters were regressed from empirical data [166] (Table 11.2). Note that although the fitted pressure ratio at zero flow rate (Pr_0) is negative, the non-surge regime is to the right side of the peak, and the parameter does not have physical meaning as modeled [118, 166]. Compressor curves with these parameters are plotted for increasing speeds in Figure 11.3. The first value of W_0 is used for the first compressor (the only compressor in the case of single-stage compression), and the second value is used for the second compressor in the two-stage, intercooled compressor configuration.

Table 11.2: Compressor model parameters.

Parameter	Physical Significance	Value
Pr_0	Pressure ratio at zero flow	-22.0
H_0	Reference semiheight of compressor curve	14.0
W_0	Reference semiwidth of compressor curve	0.007785, 0.00213991

Using the modeling framework presented in Section 11.2, the MHEX is discretized into 50 segments, for which enthalpies are established and the pseudo-transient energy balances, given by (11.7)–(11.8), are used to compute temperature. The pressure drops across the process MHEX are assumed to be 5 bar for the natural gas stream, 4 bar for the hot refrigerant stream, and 1 bar for the cold refrigerant stream. Additionally, a 0.1 bar pressure drop is assumed for each SW cooler. To facilitate satisfying the constraint of maintaining constant heat exchanger area for various concentrations of natural gas feed, the heat exchanger area A is fixed to a constant set point, A^{sp}, using an integral-only controller, as in (4.15) The controller eliminates a degree of freedom from the flowsheet model, which I select as the outlet pressure of the compressor, and replaces it with a new decision variable, A^{sp}

$$\tau_I \frac{A^{sp}}{P_{out}^0} \frac{dP_{out}}{d\hat{t}} = A - A^{sp} \tag{11.27}$$

$$P_{out}(\hat{t} = 0) = P_{out}^0 \tag{11.28}$$

Here τ_I is the integral time constant of the controller, which should be selected such that the dynamics of the controller are at least as slow as the dominant dynamic mode of the pseudo-transient MHEX model [229]. The sign of the gain between the controlled variable (A) and the selected manipulated variable must also be known (at least locally) in order to maintain closed-loop stability. A PI control law could be implemented with some potential convergence benefits, but stability of the resulting closed-loop dynamical system is more difficult to ensure.

11.3.2 Base Case Optimization Results

The process flowsheet was optimized separately for both objective functions ($OBJ1$ and $OBJ2$) and both compression sequences using the pseudo-transient, time-relaxation-based algorithm. The problems were all solved with the NLPSQP (sequential quadratic programming) solver in gPROMS [204] 5.0.1 using a 64-bit Windows 7 desktop system with a 3.40 GHz Intel Core i7 processor and 16 GB of RAM. Several randomly generated initial guesses were provided for each optimization to find the best locally optimal solution for each case. The problem formulations are described in Table 11.3, and the results of all optimization calculations are presented in Table 11.4.

Table 11.3: PRICO process optimization problem formulations.

	P1a	P1b	P2a	P2b	P3a	P3b	P4a	P4b
# of Compressors	1				2			
Compressor Model	-		Fixed η_c		-		Equal Pr	
Objective Function	$OBJ1$	$OBJ2$	$OBJ1$	$OBJ2$	$OBJ1$	$OBJ2$	$OBJ1$	$OBJ2$
CPU Time (s)	719	204	413	89	460	347	393	232

Although the same mathematical models were used in every optimization, it was found that minimization of $OBJ2$ instead of $OBJ1$ always results in a smaller MHEX, reduced mixed refrigerant flow, higher total compression ratio, and increased compressor work (Table 11.4). Optimization of $OBJ2$ also required less CPU time in every case, likely because the steeper objective function allowed local optima to be reached in fewer iterations. A comparison of optimization problems P1 and P2 reveals that the assumption of constant compressor isentropic efficiency is adequate for nominal-case flowsheet optimization (a constant $\eta_c = 80\%$ was assumed, as was used in Chapter 5), and optimization with the new compressor model results in nearly the same operating pressures, MHEX size, and mixed-refrigerant (MR) stream.

Table 11.4: PRICO process nominal-case optimization results.

	P1a	P1b	P2a	P2b	P3a	P3b	P4a	P4b
Pressure S3 (bar)	9.97	5.69	9.99	5.90	10.95	3.47	11.41	5.18
Pressure S4 (bar)	35.72	33.85	35.76	33.89	39.95	42.63	42.36	44.47
MR Flow (kmol/s)	3.889	2.838	3.889	2.889	3.823	2.010	3.777	2.376
N_2 (mol%)	16.93	13.11	16.95	13.37	18.72	10.44	19.62	13.97
CH_4 (mol%)	39.41	38.64	39.40	38.63	38.57	39.76	38.11	38.99
C_2H_6 (mol%)	27.14	26.76	27.14	26.83	27.09	25.40	27.08	25.30
C_3H_8 (mol%)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
n-C_4H_6 (mol%)	16.52	21.48	16.51	21.15	15.59	24.40	15.18	21.75
Comp. 1 D (m)	1.425	1.654	-	-	1.318	1.803	1.286	1.543
Comp. 2 D (m)	-	-	-	-	1.667	1.481	1.685	1.652
Comp. 1 Pr	3.58	5.95	3.58	5.74	2.06	4.63	1.93	2.93
Comp. 2 Pr	-	-	-	-	1.78	2.67	1.93	2.93
Total Pr	3.58	5.95	3.58	5.87	3.67	12.36	3.72	8.58
A_{MHEX} (m^2)	79871	48137	79894	49567	78123	34058	76479	38163
$OBJ1$ (MW)	**15.54**	16.05	**15.89**	16.44	**14.27**	14.82	**14.29**	14.87
$OBJ2$ (MW)	18.87	**18.44**	19.22	**18.88**	17.55	**16.73**	17.52	**16.93**

Aside from the solutions for problems P2a and P2b, every compressor operates near its nominal efficiency of 82.2% (the lowest value is 81.5%). As expected, when the compressor size is simultaneously optimized with the nominal flowsheet design, the compressor is sized such that it operates near its peak efficiency. The configuration with two compressors in sequence (with intercooling) reduces the total compression work required by the system, and it also allows for the refrigerant stream flow rate and heat exchanger size to be further reduced when heat exchanger size is penalized (P3b and P4b). Although the results found through elimination of the intermediate pressure in the two-stage compression sequence (P4) and the results with the intermediate pressure as an additional decision variable (P3) have similar objective function values, the designs are noticeably different. This is most apparent from the solution to problem P3b, where the first compressor accounts for much more of the compression work than the second, and the overall pressure ratio is also much higher.

The total refrigerant flow rates differ greatly among the designs, but the optimal refrigerant stream compositions for all design cases are relatively similar, comprising mostly

methane and ethane, and with no propane. The MHEX composite curves for the optimal designs P1a and P1b are shown in Figure 11.5. The composite curves for P2a, P3a, and P4a closesly resemble the ones for P1a, and those found in the solutions for P2b, P3b, and P4b closesly resemble the ones P1b. The designs are tightly heat-integrated, with the MHEXs found in the minimization of $OBJ1$ reaching the minimum allowable temperature driving force of 1.2 K at multiple points. Here I note the importance of using a distributed-parameter mathematical model in describing the MHEX: all MHEX designs involve reaching the minimum allowable temperature driving force at one or more interior points.

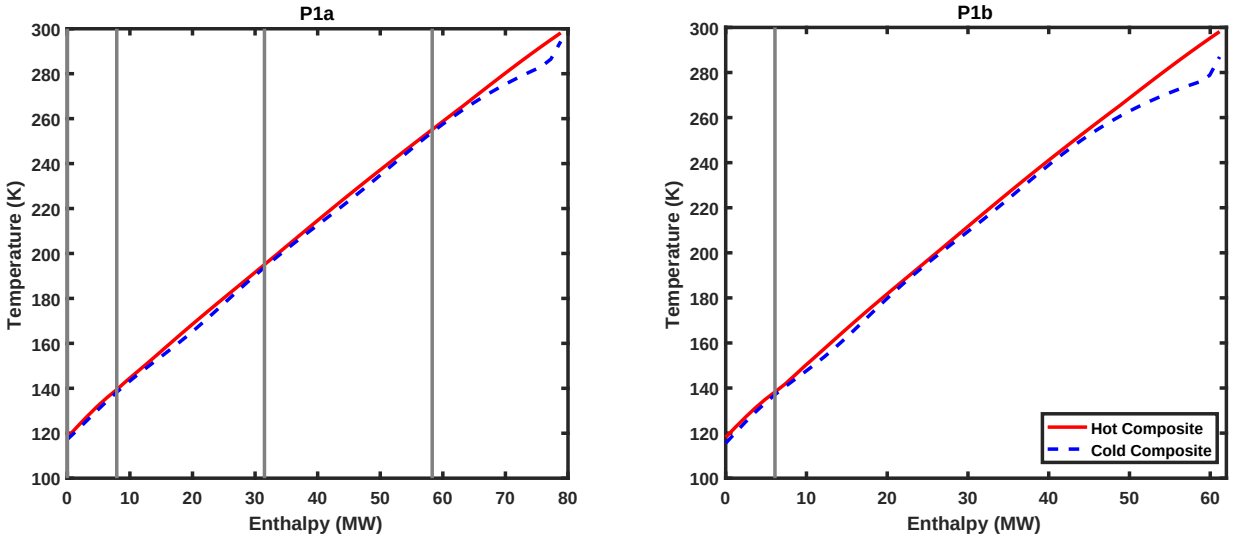

Figure 11.5: Optimal temperature-enthalpy diagram for the MHEX designs found in P1a and P2a. The vertical gray lines denote enthalpy segments where the temperature driving force exactly meets the constraint of 1.2 K.

11.3.3 Optimization Under Uncertainty Results

Upon completing the nominal-case optimization calculations presented above, the process flowsheet was optimized for an uncertain feedstock, with δ drawn from a uniform distribution $\delta \sim U(-0.1, 0.1)$ as in Table 11.1. Along with the MHEX area and the compres-

sor size(s), the refrigerant flow and composition were selected as design decision variables (**d**) for the PRICO process, such that the pressurized refrigeration cycle would not have to be altered during process operation. The rest of the decision variables were treated as recourse variables (**u**). The partitioning of decision variables into design and recourse/control variables is shown in Table 11.5.

Table 11.5: PRICO process optimization under uncertainty decision variables.

Design (**d**)	Recourse (**u**)
MR Flow	C1 Pr
MR Composition	C2 Pr
MHEX Area	
C1 Diameter	
C2 Diameter	

The problems were again solved with the NLPSQP (sequential quadratic programming) solver in gPROMS [204] 5.0.1 using a 64-bit Windows 7 desktop system with a 3.40 GHz Intel Core i7 processor and 16 GB of RAM. With the pseudo-transient smoothing of the algebraic process constraints (the process model), the maximum residual in any steady-state algebraic equation or deviation from contoller set point during integration through pseudo-time was less than 0.004%. The values of the time constants were selected to be $\tau_T = 10^1$ and $\tau_\alpha = 10^2$, and the integration horizon (t_f in (9.17)–(9.20)) was selected to be 10^6.

The optimal result for the nominal case (Table 11.4) for each problem formulation was used as the initial guess for optimization. The problem was not solved with fixed compressor efficiency (P2a and P2b), nor was the problem solved with equal compression ratios (P4a and P4b), as smaller objective function values were found by leaving the intermediate pressure as a decision variable. The results of the optimization calculations for problems P1a*, P1b*, P3a*, and P3b*, where the asterisk denotes optimization under uncertainty, are presented in Table 11.6, along with deviations of the respective values from the base case results (Δ). The solution of the optimization problems under uncertainty required slightly more memory,

with the peak memory usage in the optimizations reaching 121.23 MB, compared to 116.78 MB in the nominal case optimizations.

Table 11.6: PRICO process optimization under uncertainty results.

	P1a*	Δ	P1b*	Δ	P3a*	Δ	P3b*	Δ
MR Flow (kmol/s)	3.963	1.9%	2.976	4.6%	3.534	-7.6%	2.005	-0.3%
N_2 (mol%)	17.11	1.1%	13.83	5.5%	18.50	-1.2%	10.40	-0.4%
CH_4 (mol%)	39.41	0.0%	38.77	0.3%	38.80	0.6%	39.78	0.0%
C_2H_6 (mol%)	27.06	-0.3%	26.95	0.7%	26.70	-1.4%	25.44	0.2%
C_3H_8 (mol%)	0.00	0.0%	0.00	0.0%	0.00	0.0%	0.00	0.0%
n-C_4H_6 (mol%)	16.42	-0.6%	20.45	-4.8%	15.99	2.6%	24.37	-0.1%
Comp. 1 D (m)	1.434	0.6%	1.618	-2.2%	1.333	1.1%	1.841	2.1%
Comp. 2 D (m)	-	-	-	-	1.650	-1.0%	1.485	0.3%
A_{MHEX} (m^2)	75186	-5.9%	48970	1.7%	66173	-15.3%	34035	0.0%
E[$OBJ1$] (MW)	**15.72**	1.2%	16.05	0.0%	**14.40**	0.9%	14.83	0.0%
E[$OBJ2$] (MW)	18.92	0.3%	**18.44**	0.0%	17.30	-1.5%	**16.71**	-0.1%

The solutions for P1a*, P1b*, P3a*, and P3b* were found in 36743s, 26930s, 36030s, and 58638s of CPU time respectively. As found in Chapter 9, the bulk of the time per iteration was spent initializing the model at the initial condition of the uncertain parameter, or solving the equality constraints of the algebraic process model (8.1) using a pseudo-transient time integration. Comparably, less time was spent on the integration integration of the model (with sensitivities) through the trajectory of the uncertain parameter. While the nominal-case optimization problems were solved using a time relaxation-based algorithm, wherein each optimization iteration was initialized from the steady-state solution of the previous iteration, dynamic optimization in gPROMS does not allow for this time-saving initialization strategy. I again estimate the solution times if this capability were available by calculating the average percentage of time within each optimization iteration spent on initialization and replacing it with the amount of time required to initialize each iteration of the same nominal-case problem (since the models are approximately the same size). Initialization required approximately 81% and 91% of the total time of an iteration for the single compressor and two-stage flowsheets, respectively. Using this finding, I estimate that the

solution times for P1a*, P1b*, P3a*, and P3b* could be respectively reduced to 8429s, 5610s, 3889s, and 5635s.

The number of degrees of freedom available for MHEX operation is very limited, as is characteristic of many integrated processes [14]. In a related area, the literature surrounding the design of air separation processes demonstrates that MHEXs represent a significant design challenge when variable operation is considered, from both steady-state design [191] and dynamic operation [43] perspectives. In the results presented in Table 11.6, the optimal MHEX areas found for P1a* and P3a* are significantly reduced from those found in the nominal case optimization calculations, and the optimal refrigerant stream flow rate found for P1b* is increased. As suggested by Jensen and Skogestad [119], the exclusion of MHEX area from the objective function results in a very tightly integrated process with a large heat exchanger and refrigerant stream. While intuition might suggest that equipment should be oversized for more flexible operation, in reality the MHEX should be undersized to reduce the degree of heat integration. The temperature driving force profiles along the MHEXs in the optimal designs are plotted in Figure 11.6, where it can be seen that the designs found in P1a* and P3a* deviate slightly more from the minimum driving temperature force than the nominal case optimization results.

To compare the level of heat integration between designs, I compute the mean temperature difference $\overline{\Delta T}$ of each MHEX [119]

$$\overline{\Delta T} = \frac{1}{A} \int \Delta T dA \tag{11.29}$$

The expected value of $\overline{\Delta T}$ found in P1a* is 2.133 K, throughout the range of operation found for all possible feed concentrations, while in the nominal case, the mean temperature difference in the MHEX is 1.974 K. This confirms that the larger MHEX design is more tightly heat integrated and less inflexible, as the $\overline{\Delta T}$ is smaller than the minimum found

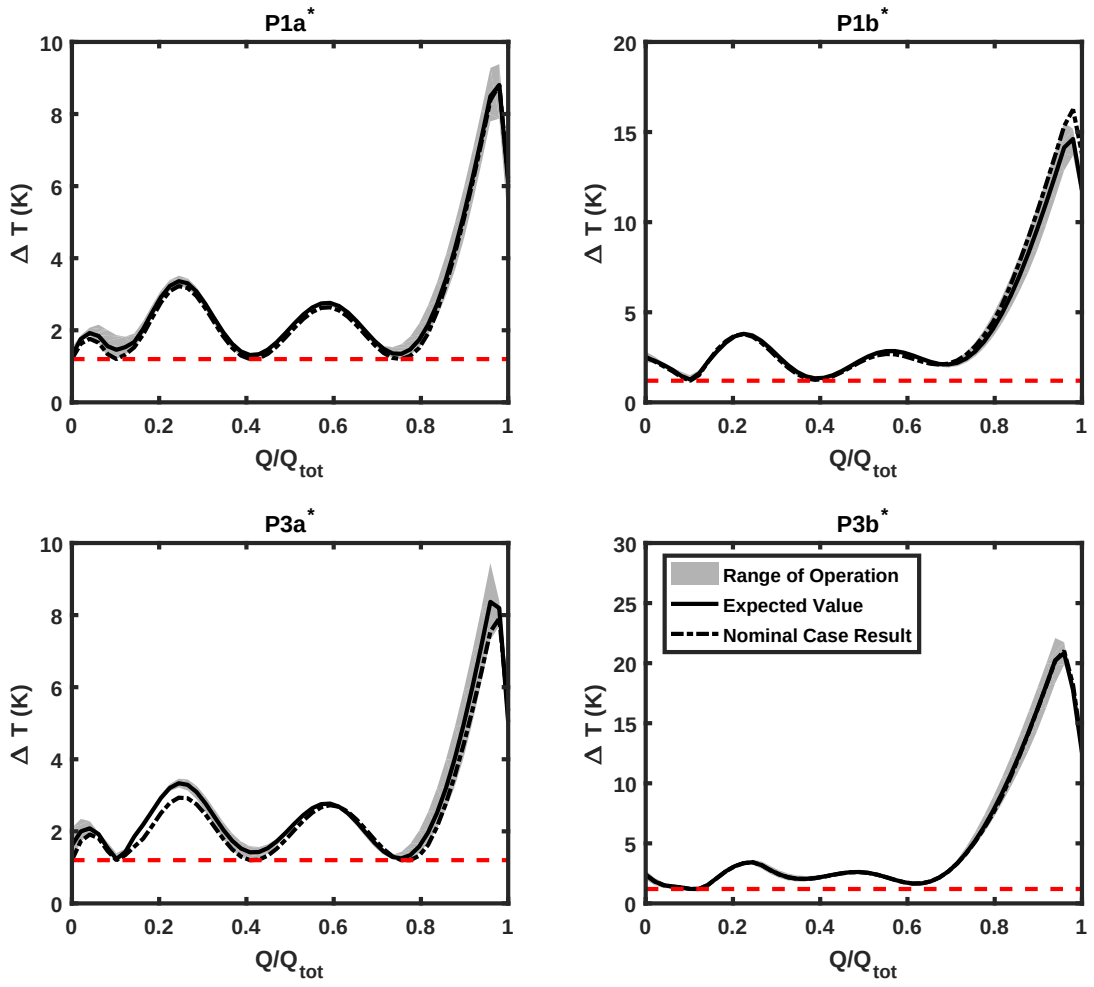

Figure 11.6: Temperature driving force along the MHEXs found by optimization under uncertainty. The shaded areas show the range of values found in the entire range of uncertain feed concentrations.

in the optimization under uncertainty case (2.131 K). The same increase in $\overline{\Delta T}$ is seen in the solution to P3a*, where the expected and minimum values are respectively 2.141 K and 2.136 K, compared to the $\overline{\Delta T}$ in the optimal nominal-case design of 1.950 K. Similarly, in the solution to P1b*, the expected and minimum values of $\overline{\Delta T}$ are 2.594 K and 2.590 K, compared to the nominal result of 2.541 K. The solution found for P3b* does not deviate

much from the nominal case optimization (P3b), and the MHEX has an identical expected $\overline{\Delta T}$ as the nominal case optimization, indicating that the design found in the solution to P3b is already flexible enough to accomodate the range of methane feed compositions considered. The required work profiles, divided between compressors, found in P3a* and P3b* are shown in Figure 11.7

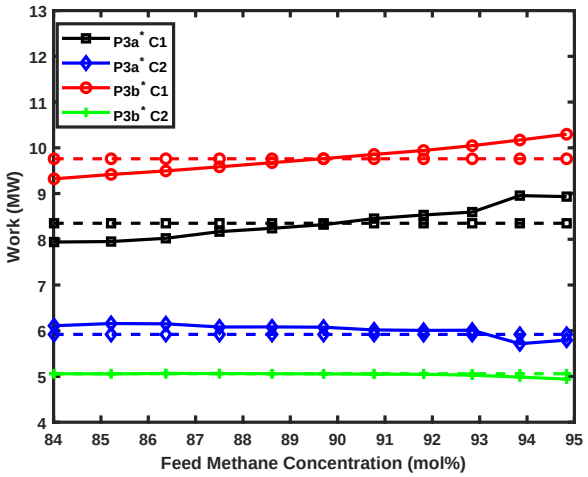

Figure 11.7: Compressor work profiles found in P3a* and P3b*. The dotted lines represent the values found in the nominal-case optimization.

As control of intensified processes require quick computation of the optimal process inputs [14], identification of the operating regimes for different feedstock concentrations along with the optimal design specifications is crucial. Therefore, the recourse variable profiles are a second important result from the solution of the optimization problem under uncertainty using the proposed sequential dynamic optimization approach. The trajectories of the recourse variables were parameterized [268] using piecewise-linear profiles, giving the optimal operating conditions for the explored range of feed conditions [259]. The profiles of the high and low pressures, along with their values found in the nominal case optimization, are shown in Figure 11.8, where it can be seen that the dependence of the optimal high and

low pressures on feed composition for various designs in nonlinear.

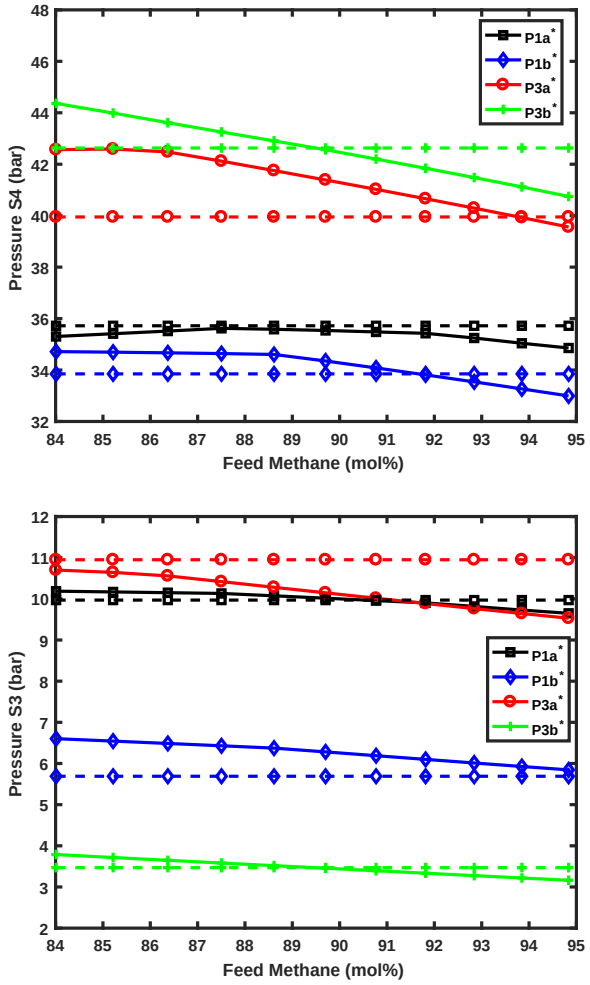

Figure 11.8: High and low pressure profiles found in the optimization under uncertainty results. The dotted lines represent the values found in the nominal-case optimization (natural gas feed at the nominal composition)

11.4 Summary

This chapter described the optimization of natural gas liquefaction processes under feedstock uncertainty. I presented mathematical models based on novel pseudo-transient continuation concepts that allow for key equipment (MHEXs and compressors) to be sized simultaneously with the flowsheet optimization. I then introduced variations in feedstock composition as a specific parametric uncertainty and optimized the representative PRICO mixed-refrigerant NG liquefaction process over a range of feed methane concentrations. In the nominal case, the use of two-stage compression allows for a significant reduction of the total required compressor power and that the inclusion of MHEX area in the objective function results in smaller, less tightly heat-integrated MHEX designs. I then optimized the same flowsheet while accounting for uncertainty in feed methane concentration, showing that, counterintuitively, the optimal MHEXs designed for the nominal-case operation (when size is not penalized) are too large–and thus too tightly heat integrated– to operate flexibly, and that they should actually be designed to be "undersized" for a feed stream of uncertain composition.

Part III

Optimization of Time-Varying Process Operations

Chapter 12

Dynamic Optimization Strategies for Process Control and Operations: Preliminaries[†]

12.1 Overview

While Part II of this dissertation focused on the optimal design of processes for operation at variable (steady-state) points, there are cases where the steady-state operating paradigm itself is sub-optimal. Optimization of process operations in these cases involves *dynamic optimization* problems, wherein the process can no longer be represented by a (steady-state) algebraic model as in (3.1) or (8.1). Rather, the dynamic behavior of the process must be considered explicitly in the optimization model. In this chapter, I review two such circumstances motivating dynamic optimization: (i) periodic process operations for *dynamic process intensification* and (ii) demand-response operation in fast-changing electricity markets. Furthermore, this chapter discusses important concepts and mathematical tools involved in the solution of dynamic optimization problems in these contexts.

[†]The contents of this chapter are largely based on the following two publications (C.T. is the primary author of both):
C. Tsay, R.C. Pattison, and M. Baldea. A pseudo-transient optimization framework for periodic processes: Pressure swing adsorption and simulated moving bed chromatography. AIChE J., 64(8):2892–2996, 2018.
C. Tsay and M. Baldea. Integrating production scheduling and process control using latent variable dynamic models. Control Eng. Pract., 94:104201, 2020.

12.2 Periodic Process and Dynamic Process Intensification

In contrast to the intensified process designs in Parts I–II, *dynamic process intensification* refers to changes in the dynamics, operation strategy, and/or control of a conventional or intensified system that leads to a substantially more efficient processing path [17]. For the case of continuous processing, dynamically intensified processes operate in a periodic/cyclical fashion, whereby a unit undergoes the same operating pattern repeatedly, or several units rotate indefinitely through a sequence of steps in time. Periodic configurations are used in many separation processes and in reaction engineering. Adsorption-based separation processes such as pressure-swing adsorption (PSA) and temperature-swing adsorption (TSA) utilize cyclical operation to exploit selective adsorption and desorption, and have garnered particular interest as an alternative to energy-intensive separation techniques such as cryogenic distillation [120, 276]. In the reaction engineering realm, one prominent example is the reverse-flow reactor, in which the same catalytic bed carries out two reactions. For example, a fuel is combusted to generate heat, which is retained by the bed. Then, the flow direction is switched, and the process stream undergoes an endothermic reaction supported by the retained heat; the process continues periodically [165]. In a different vein, the performance of many conventional reaction systems has been shown to improve through the use of periodic inputs or controls, allowing process intensification through periodic operation [236, 242].

These intensified systems generally reach and operate at a cyclic steady state (CSS) that enables reliable performance and, *on the average*, corresponds to desired production rate(s), product quality, etc. For optimization and simulation in industrial applications, the initial transient (start up) period is typically not important, and only the CSS is considered. The development of computational approaches for optimizing the design and operation of periodic processes at CSS has received attention in the recent literature [3, 126, 234]. The majority of authors indicate that the cyclic dynamics are at the origin of increased compu-

tational resource requirements and convergence issues.

Recently proposed approaches to simulate the CSS of a process have focused on formulating the dynamic (cyclic) system as an algebraic system of equations, with the temporal domain fully discretized and the CSS condition treated as an equality constraint. Several methods have been proposed to accelerate the convergence or reduce the computational expense of simulating such fully discretized systems [3, 131, 270]; nevertheless, the robust calculation of CSS in large, complex processes remains a significant computational challenge.

12.2.1 Cyclic Processes and the Cyclic Steady State (CSS) Condition

As mentioned above, only the cyclic steady state must be considered for design optimization and simulation in most industrial applications. At CSS, the process states at the end and beginning of a cycle must be the same. The process can be mathematically described as:

$$\boldsymbol{f}(\mathbf{x}, \dot{\mathbf{x}}, \mathbf{y}, \mathbf{u}, \mathbf{d}, \boldsymbol{\theta}, t) = 0 \tag{12.1}$$

$$\boldsymbol{g}(\mathbf{x}, \mathbf{y}, \mathbf{u}, \mathbf{d}, \boldsymbol{\theta}, t) = 0 \tag{12.2}$$

$$c(\mathbf{x}, t_{cycle}) = |\mathbf{x}(t) - \mathbf{x}(t + t_{cycle})| \leq tol \tag{12.3}$$

$$\mathbf{u}(t) = \mathbf{u}(t + t_{cycle}) \tag{12.4}$$

$$LB \leq \mathbf{x}, \mathbf{d} \leq UB \tag{12.5}$$

where $\boldsymbol{f}$ is the dynamic process model, $\mathbf{x} \in \mathbb{R}^n$ are the state variables, $\mathbf{y} \in \mathbb{R}^m$ are the algebraic variables, $\mathbf{u} \in \mathbb{R}^u$ are the system inputs, $\boldsymbol{\theta}$ are the model parameters, t is the physical time domain, and t_{cycle} is the duration of a full cycle. A tolerance tol is generally used as the criterion for determining whether CSS is reached. The design variables $\mathbf{d}$ and state variables $\mathbf{x}$ can be subject to lower bounds LB and upper bounds UB. In addition to the typical constraints present in dynamic optimization problems ($\boldsymbol{f}$ and $\boldsymbol{g}$), an additional

constraint c, or the "distance" from CSS, must be calculated from the values of the state variables at $t = t_{cycle}$.

12.2.2 Solution Strategies

The conventional method for identifying the CSS of a process, termed successive substitution, is to simulate the evolution of the process through time until the process behavior no longer changes from cycle to cycle. While this strategy provides a glimpse into the true dynamics of a process, this approach converges linearly to the CSS and, in many cases, thousands of cycles must be simulated before the CSS is reached [120, 183]. Nevertheless, using the successive substitution approach, Toumi et al. [250] developed an optimization scheme based on integrating a dynamic, periodic process until it reaches CSS, evaluating the objective function and constraints at the CSS, and calculating their respective derivatives with respect to the decision variables using finite difference methods. The decision variables (which may include operational decisions in $\mathbf{u}$ and/or design decisions in $\mathbf{d}$) are then updated for the subsequent optimization iteration. Several approaches have been proposed for accelerating convergence to the CSS [131]. Jiang et al. [120] implemented a direct determination method [68], in which the optimization solver simultaneously searches for the decision variables that maximize the objective function and the concentration profiles that maintain a CSS. Although the computational effort associated with time integration can be greatly reduced with this approach, the number of decision variables is greatly increased, and dynamic optimization can require millions of sensitivity evaluations.

As an alternative to dynamic optimization using time integration, Nilchan and Pantelides [183] proposed a fully discretized scheme, where the temporal dimension of a periodic process is discretized, creating a large-scale optimization problem with a set of algebraic-only constraints. In this case, CSS can be enforced as an additional algebraic constraint, and the

computational effort for time integration is completely eliminated. Instead, the computational expense is shifted to nonlinear algebraic solvers, which must handle large numbers of algebraic variables and constraints for each optimization iteration [28, 126]. This fully discretized approach, known as *simultaneous dynamic optimization*, was further improved by Kawajiri and Biegler [126], who replaced the backwards-finite-difference approximations for temporal discretization with Radau collocation on finite elements. Later, Vetukuri et al. [270] used an inexact trust-region algorithm to reduce the number of Jacobian calculations, Agarwal et al. [2] used reduced-order models to simplify the problem handled by the optimization solver, and Pattison et al. [194] used a Jacobian-free Newton-Krylov (JFNK) approach to calculate CSS conditions iteratively. Discussions on simultaneous dynamic optimization can be found in [28] and [30].

The total discretization approach enables accurate and robust solution of the dynamic behavior of a periodic process without time integration [126], but requires that the selected numerical solvers (typically Newton-type) be able to initialize, or solve all the algebraic constraints of, the process model (12.1). For optimization, the sensitivities of the objective function and constraints must also be calculated to each decision variable, increasing the required computational effort. Considering chemical process models specifically, the equations describing such systems are typically highly nonlinear, coupled, and ill-conditioned, owing to the presence of recycle streams (or periodicity conditions) and the nonlinear behavior of non-isothermal systems. Simulation of process models further increases in difficulty as spatial variations are accounted for; each included spatial domain must be discretized using finite difference or collocation methods, increasing the number of variables and the number of algebraic equations. In summary, obtaining the CSS of a cyclic/periodic process using a fully discretized approach requires a strategy that can reliably solve the large-scale, nonlinear, and highly coupled algebraic model including the CSS equality constraints.

12.3 Process Scheduling and Demand-Response Operation

The above strategies for dynamic process intensification via periodic operation are primarily enacted by the control system of a process, which may use a logic-based strategy to execute the cyclic "schedule." These control systems are responsible for maintaining the process at a (cyclic) steady state in the face of disturbances and setpoint changes. In this section, I consider the higher-level decisions that provide relevant setpoints to the control systems. Figure 12.1 illustrates the hierarchy of decisions in process operations.

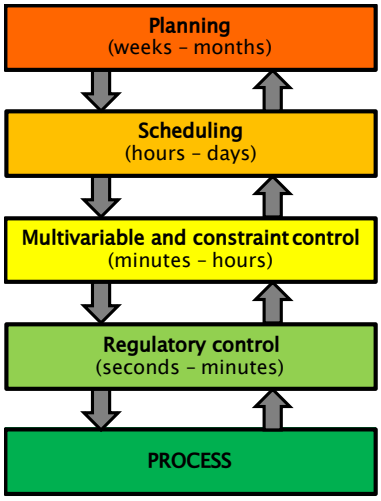

Figure 12.1: Hierarchy of decisions in process operations.

The layer directly above the process control system, *production scheduling*, determines the production sequences, product grades, batch sizes, unit assignments, and/or task timing that maximize profits (or minimize cost). In contrast the relatively fast time scales for periodic process operations, production scheduling decisions are typically made over a time horizon spanning several hours to several days. Once the optimal schedule is determined, the control layer of a process seeks to track the setpoints/targets determined by the scheduling layer, while satisfying process and product constraints. To this end, optimization-based controllers (notably model predictive control—MPC) have enjoyed widespread acceptance

in the chemical industry [206]. MPC determines optimal control moves using a dynamic model to predict the plant response (over a prediction horizon typically much shorter than the horizon of scheduling calculations) to changes in the manipulated variables.

Many recent works focus on this scheduling layer for chemical processes, due to its relevance to *demand response* (DR) operation [293]. Demand response, or "load shifting," refers to intentional modification by an electricity user of its power consumption over time in order to exploit time-dependent electricity prices. This behavior is an apparent "win-win," reducing costs for electricity users while also mitigating (temporary) mismatches between supply and demand in the power grid. The increasing prevalence of intermittent renewable energy sources introduces additional short-term and long-term (daily to seasonal) fluctuations in electricity supply. As a result, real-time electricity prices may fluctuate by several orders of magnitude during a 24-hour period, providing a strong incentive for DR [56]. Industrial manufacturing plants are a natural candidate for DR, since they may intuitively overproduce product(s) when electricity prices are low, store excess product(s), and fill orders using stored product(s) when electricity prices are high (and production rates are reduced). Participation in a real-time electricity market is largely voluntary for industrial consumers, but can become economically attractive when price variations are high, excess production capacity is available, and excess product(s) can be stored safely and at reasonable cost.

12.3.1 Integrated Scheduling and Control

Conventional methods for computing optimal schedules mostly rely on the assumption that the process is at a steady state before each change in production targets, and that it will again reach a new steady state shortly thereafter. Transition times, while accounted for, are implicitly assumed to be much shorter than the time(s) spent operating at a steady state. However, this assumption may not be valid when scheduling decisions are made

over shortened time intervals, such as those required for DR. Electricity prices in current deregulated markets may change at hourly (or faster) time intervals, and scheduling decisions must therefore be made over a time scale in which process dynamics and control become highly relevant [292]. Specifically, production targets must change over hourly or faster intervals, and process safety operating constraints must be accounted for when imposing such rapid production modulation. The decision-making time scale is further shortened in faster electricity markets (e.g., the fifteen minute market), when it is desired to provide ancillary services, or place electricity bids in real-time [73, 188, 223].

In light of these developments, the *integration of scheduling and control* for chemical processes is an important research area. Advances in this field have been the subject of several recent review papers [18, 61, 66]. The integrated scheduling and control problem aims to derive dynamically feasible production schedules by including the *closed-loop* dynamics of a process (i.e., including the control strategy) in the scheduling problem. The deterministic, continuous-time integrated scheduling and control problem can be stated generally as a dynamic optimization problem [74]:

$$\max_{\mathbf{y}^{sp}(t)} \quad \int_0^{t_f} J(\mathbf{y}, t)dt \tag{12.6}$$

$$s.t. \quad \boldsymbol{f}(\mathbf{x}, \dot{\mathbf{x}}, \mathbf{y}, \mathbf{u}, \mathbf{d}, \boldsymbol{\theta}, t) = 0 \tag{12.7}$$

$$\boldsymbol{g}(\mathbf{x}, \mathbf{y}, \mathbf{u}, \mathbf{d}, \boldsymbol{\theta}, t) = 0 \tag{12.8}$$

$$\mathbf{u} = \boldsymbol{K}(\mathbf{y}^{sp} - \mathbf{y}) \tag{12.9}$$

$$l(\mathbf{x}, \mathbf{y}, \mathbf{u}, t) = 0 \tag{12.10}$$

$$LB \leq \mathbf{x}, \mathbf{y}, \mathbf{u} \leq UB \tag{12.11}$$

where $\mathbf{y}^{sp}(t)$ is a time-varying vector of production targets and/or other setpoints to be supplied to the control system, and $\boldsymbol{K}$ represents the process control policy. The process dynamics are again represented by $\boldsymbol{f}$ and $\boldsymbol{g}$. The process variables $\mathbf{x} \in \mathbb{R}^n, \mathbf{y} \in \mathbb{R}^m, \mathbf{u} \in$

$\mathbb{R}^u$ may be subject to constraints during optimization (12.11). The economic objective function $J(\mathbf{y}, t)$ typically includes the revenue from selling product, the process operating costs, penalties for storage, etc. $l(\cdot)$ includes storage and demand constraints that ensure (i) customer demand can be met at all times, (ii) the amount of product stored does not deplete/exceed the physical capacity of the storage system, and (iii) artificial economic gains are not realized by depleting hold-up present at $t = 0$. Note that $l(\cdot)$ may include both path ($0 \leq t \leq t_f$) and endpoint ($t = t_f$) constraints.

12.3.2 Solution Strategies

In an early effort to integrate process scheduling and control, Flores-Tlacuahuac and Grossmann [90] explicitly included the dynamic process model and controller in the scheduling problem, resulting in a large simultaneous dynamic optimization problem. Zhuge and Ierapetritou [304] later implemented this discretized-time approach in a closed-loop strategy to mitigate the effect of disturbances. Beal et al. [22] extended this concept to account for time-dependent parameters and constraints, and demonstrated the economic benefits of integrated scheduling control in both open-loop and closed-loop implementations [23]. Koller et al. [138] considered embedding PI controllers into scheduling calculations, accounting for stochastic disturbances and uncertainties using a sample-based, back-off method.

In general, embedding a dynamic process model in scheduling calculations tends to increase computational cost, and many optimization techniques have been proposed to facilitate dealing with the integrated problem. Nyström et al. [185] reduced the computational complexity of solving the integrated problem by decomposing it into a scheduling master problem and a control sub-problem, and iterating between the two. Nie et al. [181] took a similar approach, proposing a generalized Benders decomposition algorithm, where the scheduling decisions comprise the master problem and the dynamic process optimization

comprises the primal problem. Simkoff and Baldea [237] directly incorporated the KKT optimality conditions of a linear MPC system using complementary constraints to provide an exact representation of closed-loop dynamics in the scheduling layer.

Several works (e.g., [41, 305]) employed multiparametric model predictive control (mpMPC) in an optimal scheduling framework. Charitopoulos et al. [52] examined the closed-loop implementation of an mpMPC approach for integrated scheduling and control that can handle dynamic disturbances. mpMPC approaches rely on generating explicit forms of the optimal control laws offline, and are thus computationally efficient when implemented online. However, the complexity of the offline problem grows exponentially with the size of the system (model) under consideration and with the dimension of its input and output vectors.

Chapter 13

A Pseudo-Transient Framework for Modeling, Simulation, and Optimization of Periodic Processes[†]

Part I of this dissertation presented a robust approach for the modeling, simulation, and optimization of detailed (spatially distributed) process models based on pseudo-transient continuation, referring strictly to *steady-state* systems. In this chapter, I formulate a new set of pseudo-transient simulation and optimization principles for *dynamic* process models, based on discretizing the temporal domain similarly to (and in addition to) spatial domains, thereby creating an algebraic system of equations. This latter system is then subjected to a pseudo-transient reformulation, for which I provide some theoretical stability guarantees. Motivated by applications with periodicity (i.e., cyclic steady state) constraints, I establish a fundamental structural analogy between the cyclic steady state condition and the constraints used to represent material/energy recycling in flowsheet models. I note that the CSS condition amounts to "recycling in the time domain," and exploit this property to break the algebraic loops introduced by CSS equations via a dynamic "tearing-like" procedure. These concepts further address the challenges described in Section 2.6 pertaining to *complex, physical models*. The presentation in this chapter follows closely the material published in Tsay et al. (2018) [261].

[†]The contents of this chapter are largely based on the following publication: C. Tsay, R.C. Pattison, and M. Baldea. A pseudo-transient optimization framework for periodic processes: Pressure swing adsorption and simulated moving bed chromatography. AIChE J., 64(8):2892–2996, 2018. C.T. is the primary author of the manuscript.

13.1 Pseudo-Transient Formulation for a Discretized Time Domain

This section presents the mathematical details of the proposed formulation for pseudo-transient simulation and optimization of fully discretized dynamic process models. I begin with the proposed scheme for handling time derivatives in a discretized temporal domain, followed by stability considerations and application of dynamic "tearing" to handle CSS conditions. I note that these dynamic models are consistently formulated using the pseudo-transient approach and can be seamlessly integrated with a previously developed library of unit operations [190] and detailed process unit models (as discussed in Chapters 4–6).

13.1.1 Modulation of Time Derivatives

Consider the pseudo-transient model of a dynamic system (12.1), where the evolution of the state variables can be expressed as a set of explicit first-order ODEs ($\boldsymbol{\phi}$) with the corresponding initial conditions:

$$\frac{\partial \mathbf{x}}{\partial t} = \boldsymbol{\phi}(\mathbf{x}, \mathbf{y}, \mathbf{u}, \mathbf{d}, \boldsymbol{\theta}, t) \tag{13.1}$$

$$\mathbf{x}(t=0) = \mathbf{x}_0 \tag{13.2}$$

where t represents the physical time domain (recall that $\hat{t}$ denotes the pseudo time domain and has no physical meaning). Further, I consider a time-discretized system, wherein the temporal domain is discretized into N points. The time derivative of a state variable x can then be written as an N-dimensional vector, where each entry corresponds to a time point:

$$\dot{x}_1 = \phi(x_1, \mathbf{y}_1, \mathbf{u}_1, \mathbf{d}, \boldsymbol{\theta}, t_1) \tag{13.3}$$

$$\dot{x}_2 = \phi(x_2, \mathbf{y}_2, \mathbf{u}_2, \mathbf{d}, \boldsymbol{\theta}, t_2) \tag{13.4}$$

$$\vdots$$

$$\dot{x}_N = \phi(x_N, \mathbf{y}_N, \mathbf{u}_N, \mathbf{d}, \boldsymbol{\theta}, t_N) \tag{13.5}$$

and the spacing of the time points $t_1, ..., t_N$ depends on the discretization scheme employed. $\dot{x}_i$ denotes the algebraic equation(s), i.e. functions of x_i, representing the time derivative of x at time t_i in the selected discretization scheme. I now reformulate these dynamics using the same principles as those applied to the modulation of source/sink terms in spatial dimensions (3.11)–(3.3.2), imposing no temporal variations at the consistent initialization step and slowly introducing the temporal variations as the system is integrated through pseudo-time until they are fully present at steady state. This modulation is accomplished by introducing a new variable $\hat{\boldsymbol{\phi}}$ such that:

$$\frac{\partial \mathbf{x}}{\partial t} = \hat{\boldsymbol{\phi}}(t) \tag{13.6}$$

$$\mathbf{x}(t=0) = \mathbf{x}_0 \tag{13.7}$$

Written in discretized form for a single example state variable x:

$$\dot{x}_1 = \hat{\phi}_1 \tag{13.8}$$

$$\dot{x}_2 = \hat{\phi}_2 \tag{13.9}$$

$$\vdots$$

$$\dot{x}_N = \hat{\phi}_N \tag{13.10}$$

The pseudo-time dynamics of the new variables $\hat{\boldsymbol{\phi}}$ are defined such that the solution to the original algebraic system, i.e. the discretized ODEs (13.3)-(13.5), is recovered at steady state:

$$\tau_\phi \frac{d\hat{\boldsymbol{\phi}}_i}{d\hat{t}} = \boldsymbol{\phi}(\mathbf{x}_i, \mathbf{y}_i, \mathbf{u}_i, \mathbf{d}, \boldsymbol{\theta}, t_i) - \hat{\boldsymbol{\phi}}_i \tag{13.11}$$

$$\hat{\boldsymbol{\phi}}_i(\hat{t}=0) = \mathbf{0} \tag{13.12}$$

$$\forall\ i = 1, 2, ..., N$$

The algebraic system resulting from the discretization of the time domain requires that the entire trajectory of x, given by (13.8)-(13.10), be solved at every integration step in pseudo-time as it converges to the solution to the original algebraic system (13.3)-(13.5) as $\hat{t} \to \infty$. The solution of the original state variable x trajectory in time at $\hat{t} = 0$ (consistent initialization of the pseudo-transient model) is simplified to $x_1, x_2, ..., x_N = x_0$, the initial condition for x. I establish the stability of this proposed scheme in the following theorem.

Theorem 13.1.1. *For a function $\boldsymbol{\phi}$ that is continuous and has Lipschitz continuous first and second derivatives, the proposed pseudo-transient formulation in* (13.11)-(13.12) *is asymptotically stable to perturbations around the solution to the original algebraic system* (13.3)-(13.5)*, written as $\dot{\mathbf{x}}_i|_{\mathbf{x}^*} = \boldsymbol{\phi}(\mathbf{x}_i^*, \mathbf{y}_i, \mathbf{u}_i, \mathbf{d}, \boldsymbol{\theta}, t_i) = \boldsymbol{\phi}^*, \forall\ i = 1, ..., N$ given that an appropriate time discretization step Δt is chosen.*

Proof. To assess the dynamic stability in pseudo time of the variable $\hat{\boldsymbol{\phi}}$, I perform a linear analysis of the first order dynamics of a single variable $\hat{\phi}$, given by (13.11). At the pseudo-transient steady state (algebraic system solution) of (13.11)-(13.12), the system is described by:

$$\tau_\phi \frac{d\hat{\phi}_i}{d\hat{t}}\bigg|_{x_i^*, \hat{\phi}_i^*} = \mathcal{F}(x^*, \hat{\phi}*) := \phi(x_i^*, \mathbf{y}, \mathbf{u}, \mathbf{d}, \boldsymbol{\theta}, t_i) - \hat{\phi}_i^* = 0 \tag{13.13}$$

where x_i^* is the true solution of the discretized algebraic system at time t_i, and $\hat{\phi}_i^*$ is the derivative of x_i^* with respect to t according to the system equations (13.1). Beginning with the initial condition $x(t = 0) = x_0$, I express the dynamics of $\hat{\phi}_0 = \hat{\phi}(t = 0)$ as:

$$\tau_\phi \frac{d\hat{\phi}_0}{d\hat{t}} = \mathcal{F}_0(x, \hat{\phi}) := \phi(x_0) - \hat{\phi}_0 \tag{13.14}$$

For a small perturbation η_0 around the solution $\hat{\phi}_0^*$, it can be assumed that the

peturbation affects the system linearly:

$$\mathcal{F}_0(x_0^*, \hat{\phi}_0^* + \eta_0) \approx \frac{\partial \mathcal{F}_0}{\partial \hat{\phi}_0}\eta_0 \tag{13.15}$$

$$\frac{\partial \mathcal{F}_0}{\partial \hat{\phi}_0} = \frac{\partial}{\partial \hat{\phi}_0}\phi(x_0) - 1 = -1 \tag{13.16}$$

where the value of $\mathcal{F}_0(x_0^*, \hat{\phi}_0^*)$ is known to be zero (13.13) and omitted. A perturbation of a linear system decays in time if the derivative of the governing dynamic equation with respect to the perturbed variable is negative [6]. Thus the perturbation η_0 decays exponentially in pseudo-time, and the state $\hat{\phi}_0^*$ is stable. In a similar way, I now consider the first discretized time point $\hat{\phi}(t = \Delta t) = \hat{\phi}_1$, for which the dynamics can be expressed as:

$$\tau_\phi \frac{d\hat{\phi}_1}{d\hat{t}} = \mathcal{F}_1(x, \hat{\phi}) := \phi(x_1) - \hat{\phi}_1 \tag{13.17}$$

where x_1 is the value of x at time $t = \Delta t$. Proceeding with the same linear analysis, I write x_1 as $x_1 \approx x_0 + \hat{\phi}_0 \Delta t$. The dynamics of $\hat{\phi}_1$ can thus be expressed:

$$\tau_\phi \frac{d\hat{\phi}_1}{d\hat{t}} = \mathcal{F}_1(x, \hat{\phi}) = \phi(x_0 + \hat{\phi}_0\Delta t) - \hat{\phi}_1 \approx \phi(x_0) + \left.\frac{\partial \phi}{\partial x}\right|_{x_0} \hat{\phi}_0 \Delta t - \hat{\phi}_1 \tag{13.18}$$

Consider again a perturbation η_1 around the solution $\hat{\phi}_1^*$, assuming the system can be linearly approximated around its steady-state solution:

$$\mathcal{F}_1(x_1^*, \hat{\phi}_1^* + \eta_1) \approx \frac{\partial \mathcal{F}_1}{\partial \hat{\phi}_1}\eta_1 \tag{13.19}$$

$$\frac{\partial \mathcal{F}_1}{\partial \hat{\phi}_1} = \left.\frac{\partial^2 \phi}{\partial \hat{\phi} \partial x}\right|_{x_0} \hat{\phi}_0 \Delta t - 1 \tag{13.20}$$

It can be seen from (13.20) that the derivative $\frac{\partial F_1}{\partial \hat{\phi}_1}$ is made negative if a small enough value of temporal discretization Δt is used (the function ϕ has Lipschitz continuous first and second derivatives). Behavior of the perturbation η_1 is therefore dependent on the coarseness of discretization, and the state $\hat{\phi}_1^*$ is stable if a small enough time step Δt is selected. The

stability of the system at the rest of the discretized time points can be evaluated following a similar analysis as (13.18), written as:

$$\tau_\phi \frac{\partial \hat{\phi}_i}{\partial \hat{t}} = \mathcal{F}_i(x, \hat{\phi}) \approx \phi(x_{i-1}) + \left.\frac{\partial \phi}{\partial x}\right|_{x_{i-1}} \hat{\phi}_{i-1}\Delta t - \hat{\phi}_i \tag{13.21}$$

$$\frac{\partial \mathcal{F}_i}{\partial \hat{\phi}_i} = \frac{\partial}{\partial \hat{\phi}_i}\phi(x_{i-1}) + \left.\frac{\partial^2 \phi}{\partial \hat{\phi} \partial x}\right|_{x_{i-1}} \hat{\phi}_{x-1}\Delta t - 1 \tag{13.22}$$
$$\forall\ i = 1, 2, ..., N$$

Furthermore, (13.20) and (13.22) are equivalent if $\frac{\partial}{\partial \hat{\phi}_i}\phi(x_{i-1})$ is zero, a condition that is satisfied if (13.3)-(13.5) are solved, or the solution to the original system has been recovered for previous discretization points, such that $\phi(x_{i-1}) = \hat{\phi}^*_{i-1}$. Consequently, the proposed pseudo transient formulation of the fully discretized system will be stable for some region around the solution $\hat{\phi}^*$ when the points $(t_1, ..., t_N)$ are solved sequentially, from the initial condition to the end of the discretized temporal domain provided that $\Delta t = t_i - t_{i-1}$ is sufficiently small. I note that the dynamic behavior of many chemical processes can largely be described by transitions from one state to another, shown in Figure 13.1, and thus the state variables at earlier time points in the transition will not have to move as far from the initial conditions compared to those at later time points. In the case of such systems, it is natural to therefore expect that the time discretization points will reach steady state sequentially, allowing the system to maintain a stable integration path in pseudo time.

This pseudo-transient formulation for dynamic systems can be easily extended to spatially distributed dynamic models. Discretization of the spatial domain(s) of a process model, e.g. using the method of lines [224], results in the conversion from a PDE system to an ODE system of multiple sets of state variables (and additional equality constraints from the discretization method), each corresponding to a point along the discretized spatial dimension. Thus, unless mesh refinement is required, the application of the proposed pseudo-

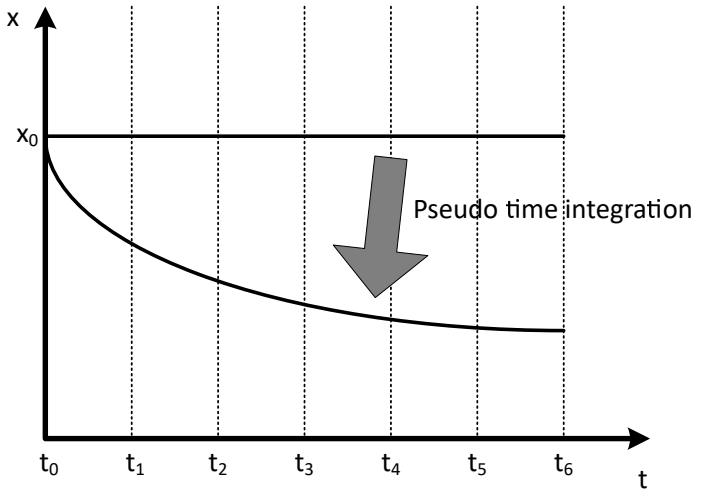

Figure 13.1: Sample state variable profile from consistent initialization (flat profile) to the trajectory that represents the steady-state solution of the pseudo-transient system.

transient formulation to a fully discretized distributed system is a trivial extension of the concepts presented above. □

13.1.2 Cyclic/Periodic Processes

In this chapter, I treat the temporal dimension of a dynamic process by total discretization (similar to the treatment of spatial dimensions in Chapters 4–6) thereby creating a large-scale algebraic system, with any modeled spatial dimensions also discretized. The CSS conditions (12.3) for a cyclic process are given as a set of system conditions at particular times that must be equivalent and are thus coupled "boundary conditions" in the time domain that must be met. The entire algebraic system is then modeled as a pseudo-transient, DAE system. As described in (3.12), equality constraints resulting in coupled algebraic systems (algebraic loops) can be treated using a first-order filter, or dynamic tearing procedure in pseudo-time. Thus, I propose to decouple the CSS equations of a process at the consistent initialization step by applying (3.12), which was previously developed to address the similar mathematical challenge of decoupling material and energy recycle loops. Figure 13.2 shows

a general cyclic process with a single spatial dimension modeled, in which the conditions along an axial dimension at the beginning of each cycle ($t = 0$) must match the conditions at the end of a cycle ($t = t_{cycle}$).

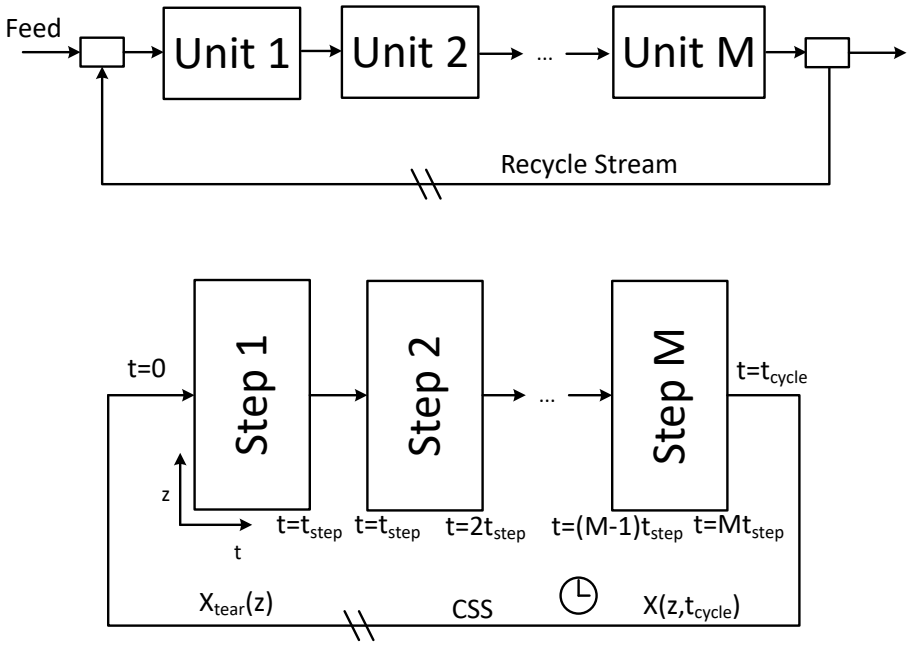

Figure 13.2: Conceptual comparison between a steady-state recycle stream (top) and a dynamic CSS condition (bottom). The CSS condition is shown here for a process model with one spatial dimension z and can be viewed as a "recycle" loop in the temporal domain.

The general CSS condition for this example process (12.3) is converted to a pseudo-transient dynamic tear by applying a first-order filter (3.12), expressed as follows for a PDE system with a single spatial dimension z:

$$\mathbf{X}(z,0) = \mathbf{X}_{tear}(z) \tag{13.23}$$

$$\tau_r \frac{\partial \mathbf{X}_{tear}(z)}{\partial \hat{t}} = \mathbf{X}(z,t_{cycle}) - \mathbf{X}_{tear}(z) \tag{13.24}$$

where $\mathbf{X}$ is the vector of state variables to be matched at CSS, generally including compositions, temperatures, etc. If the initial guess profile $\mathbf{X}_{tear}(\hat{t} = 0)$ is close to CSS operation, as shown in Figure 13.3, I expect that the dynamic system will be stable, per Theorem

13.1.1. A "flat" profile, such as $\mathbf{X}_{tear}(z) = \mathbf{X}_{in}$, could be used as an initial guess, or alternatively, the dynamic system can be simulated once (without CSS), and the dynamics of the first cycle be used to initialize the pseudo-transient model. Initialization from a simulation of a complete cycle guarantees that all of the algebraic constraints, corresponding to the mathematical process model, will be satisfied, and the CSS condition can be obtained by integrating (13.23)-(13.24) through pseudo time to steady state.

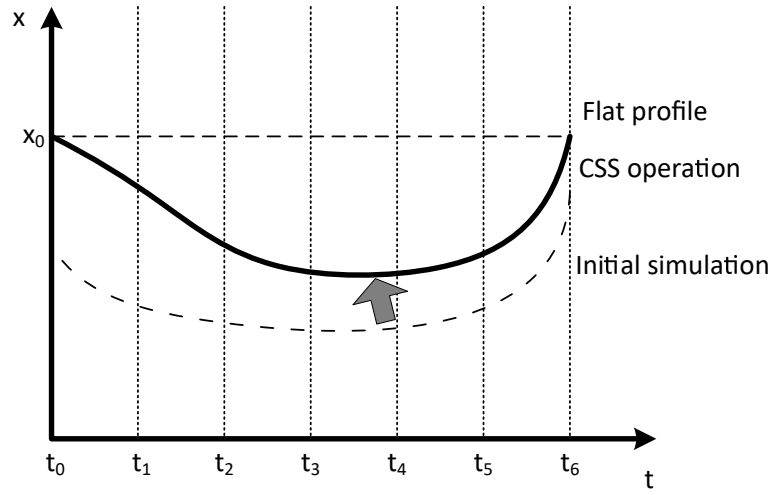

Figure 13.3: Sample state variable profile from consistent initialization to cyclic steady state (CSS), starting from either a flat profile (initial guess based on (13.11)-(13.12)) or a dynamic simulation (dashed-dotted line).

With a cyclic system, it is less intuitive that the values of the state variables at every time point can be solved sequentially from an initial guess, which would guarantee asymptotic stability of the pseudo-transient system. To this end, it may be advantageous to exploit the structured, ordered nature of a discretized model to solve the time-discretized points by "sweeping" through them chronologically, rather than solving all the time points simultaneously. Specifically, I propose to artificially enforce a sequential solution of time discretization points by making the value of the time constant τ_ϕ a function of location in

the discretized time domain:

$$\tau_\phi(t_i) = \tau_{\phi,0}\exp(\gamma\frac{t_i}{t_{cycle}}), \forall\ i = 1, ..., N \tag{13.25}$$

By treating τ_ϕ as a function of the physical time domain, the pseudo-time dynamics near the earlier time discretization points can be made faster than those at the later time discretization points (the magnitude of this difference is determined by a new parameter γ), allowing the time points to be solved sequentially (Figure 13.1).

13.2 Case Study 1: Operational Optimization of Simulated Moving Bed Chromatography

Simulated moving bed (SMB) chromatography is used for important separations in sugar, petrochemical, and pharmaceutical processes. The underlying concept of SMB processes is a countercurrent movement between liquid and solid phases, with solid "flow" simulated by cycling the locations of the inlet and outlet ports periodically [297]. A typical SMB system consists of multiple columns connected in a cycle, with ports for feed, desorbent, extract, and raffinate streams, shown in Figure 13.4.

After a preset period of time, the locations of the inlet and outlet ports are shifted forward (clockwise in Figure 13.4) by one column. Because of this controlled cycling, the SMB system eventually reaches a cyclic steady state, where the profiles of both the liquid and solid phases at the beginning of a cycle are identicaly to those at the end of the cycle, when the inlet and outlet port locations are switched. The operating parameters for industrial SMB processes are generally determined by trial-and-error and/or heuristic operational optimization [126]. In this case study, I adopt the dynamic model given by Kawajiri and Biegler [126] for a SMB 1,1'-bi-2 naphthol enantiomeric separation process, in which radial variations in the columns, thermal effects, and pressure losses are assumed to be negligible.

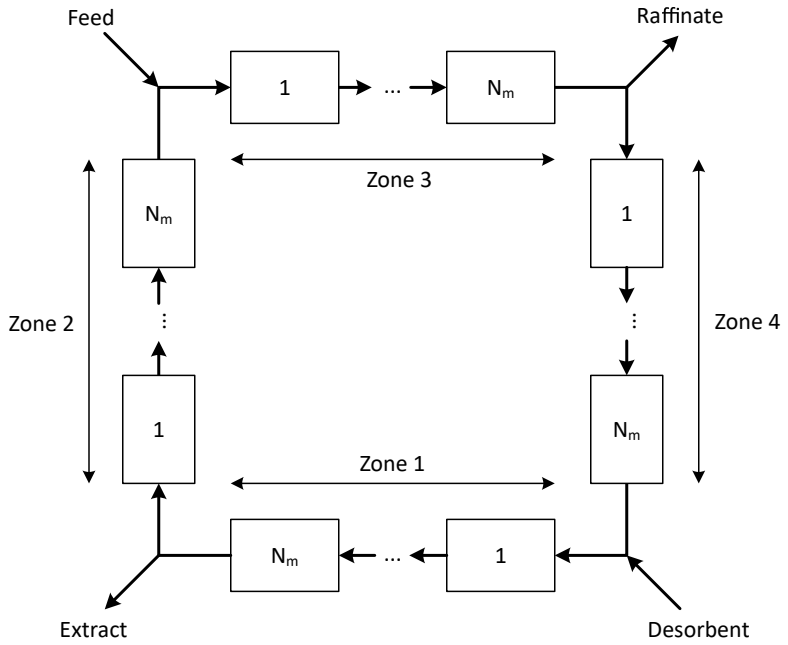

Figure 13.4: Typical SMB process with N_m columns in each zone.

Linear driving force models are used, and the mass balances in the liquid phase and in the solid-phase adsorbent are given by:

$$\epsilon_b \frac{\partial C_{n,i}(z,t)}{\partial t} + (1-\epsilon_b)\frac{\partial q_{n,i}(z,t)}{\partial t} + v_m(t)\frac{\partial C_{n,i}(z,t)}{\partial z} = 0 \tag{13.26}$$

$$\frac{\partial q_{n,i}(z,t)}{\partial t} = K_{apps,i}(q^{eq}_{n,i}(z,t) - q_{n,i}(z,t)) \tag{13.27}$$

$$q^{eq}_{n,i}(z,t) = f(C_{n,i}(z,t)) \tag{13.28}$$

$$\forall\ n = 1,2,...N_{Column}, \quad m = 1,2,3,4 \tag{13.29}$$

where ϵ_b is the void fraction, $C_{n,i}$ is the liquid-phase concentration of component i in column n, $q_{n,i}$ is the solid-phase concentration, $q^{eq}_{n,i}$ is the equilibrium solid-phase concentration, v_m is the superficial liquid velocity in zone m, and $K_{apps,i}$ is the solid-phase based mass transfer coefficient. T he total number of columns N_{Column} is equal to 4 times the number of columns in each zone N_m.

The equilibrium solid-phase concentration is given by the bi-Langmuir isotherm [167] to account for adsorption saturation and interactions between the two components:

$$q_{n,i}^{eq}(z,t) = \frac{\mathbf{H}_{1,i} C_{n,i}(z,t)}{1 + \mathcal{K}_{1,1} C_{n,1}(z,t) + \mathcal{K}_{1,2} C_{n,2}(z,t)} + \frac{\mathbf{H}_{2,i} C_{n,i}(z,t)}{1 + \mathcal{K}_{2,1} C_{n,1}(z,t) + \mathcal{K}_{2,2} C_{n,2}(z,t)} \quad (13.30)$$

$$\mathbf{H} = \begin{bmatrix} 3.73 \text{ g/l} & 2.69 \text{ g/l} \\ 0.30 \text{ g/l} & 0.10 \text{ g/l} \end{bmatrix} \quad (13.31)$$

$$\mathcal{K} = \begin{bmatrix} 0.0336 & 0.0466 \\ 1.0 & 3.0 \end{bmatrix} \quad (13.32)$$

The superficial liquid velocity in each zone is treated as a design decision variable, and the velocities of the feed, raffinate, extract, and desorbent streams (v_F, v_R, v_E, and v_D respectively) are calculated from mass balances at each inlet and outlet port:

$$v_4 + v_D = v_1 \quad (13.33)$$

$$v_1 - v_E = v_2 \quad (13.34)$$

$$v_2 + v_F = v_3 \quad (13.35)$$

$$v_3 - v_R = v_4 \quad (13.36)$$

The boundary conditions in the spatial dimension can also be calculated from mass balances at each inlet and outlet port:

$$C_{1,i}(0,t) v_1 = C_{N_{Column},i}(L,t) v_4 \quad (13.37)$$

$$C_{N_m+1,i}(0,t) = C_{N_m,i}(L,t) \quad (13.38)$$

$$C_{2N_m+1,i}(0,t) v_3 = C_{2N_m,i}(L,t) v_2 + C_{feed,i} v_F \quad (13.39)$$

$$C_{3N_m+1,i}(0,t) = C_{3N_m,i}(L,t) \quad (13.40)$$

where $C_{feed,i}$ is the concentration of component i in the feed stream. Within each zone, the boundary conditions for each column after the first are set such that the inlet concentrations

of each column are equal to the outlet concentrations of the previous column (in order of connection). For the process to be at CSS, the spatial concentration profiles in all columns at the end of a cycle must be identical to those at the beginning of a cycle. Equivalently, I define a *step* as the period of time before the inlet and outlet ports are rotated and observe that the state variable profiles within each column at the beginning of a step must be identical to those of the downstream adjacent column at the end of a step. The boundary conditions that must be satisfied in the temporal domain (CSS conditions) are thus:

$$C_{n,i}(z,0) = C_{n+1,i}(z,t_{step}), \forall\ n = 1,...,N_{Column}-1 \tag{13.41}$$

$$C_{N_{Column}}(z,0) = C_{1,i}(z,t_{step}) \tag{13.42}$$

$$q_{n,i}(z,0) = q_{n+1,i}(z,t_{step}), \forall\ n = 1,...,N_{Column}-1 \tag{13.43}$$

$$q_{N_{Column}}(z,0) = q_{1,i}(z,t_{step}) \tag{13.44}$$

13.2.1 Pseudo-Transient Formulation

Following the proposed pseudo-transient reformulation, the temporal domain is first discretized. Two new vectors of state variables $\hat{\phi}_C(z,t)$ and $\hat{\phi}_q(z,t)$ are defined such that:

$$\frac{\partial C_{n,i}(z,t)}{\partial t} = \hat{\phi}_C(z,t) \tag{13.45}$$

$$\frac{\partial q_{n,i}(z,t)}{\partial t} = \hat{\phi}_q(z,t) \tag{13.46}$$

at all discretized time points. The mass balance equations (13.26)-(13.29) are then reformulated as pseudo-transient equations:

$$\tau_{\phi_C}\frac{\partial\hat{\phi}_C(z,t)}{\partial\hat{t}} = \frac{-1}{\epsilon_b}\left((1-\epsilon_b)\frac{\partial q_{n,i}(z,t)}{\partial t} + v_m(t)\frac{\partial C_{n,i}(z,t)}{\partial z}\right) - \hat{\phi}_C(z,t) \tag{13.47}$$

$$\tau_{\phi_q}\frac{\partial\hat{\phi}_q(z,t)}{\partial\hat{t}} = K_{apps,i}(q^{eq}_{n,i}(z,t) - q_{n,i}(z,t)) - \hat{\phi}_q(z,t) \tag{13.48}$$

$$\hat{\phi}_C(\hat{t}=0) = \hat{\phi}_q(\hat{t}=0) = 0 \tag{13.49}$$

The CSS condition (13.41)-(13.44) is reformulated using the dynamic tearing procedure:

$$C_{N_{Column}}(z,0) = C_{tear}(z) \tag{13.50}$$

$$q_{N_{Column}}(z,0) = q_{tear}(z) \tag{13.51}$$

$$\tau_r \frac{\partial C_{tear}(z)}{\partial \hat{t}} = C_{1,i}(z, t_{step}) - C_{tear}(z) \tag{13.52}$$

$$\tau_r \frac{\partial q_{tear}(z)}{\partial \hat{t}} = q_{1,i}(z, t_{step}) - q_{tear}(z) \tag{13.53}$$

The parameters for the physical system are provided in Table 13.1.

Table 13.1: Parameter values for SMB process [126].

Parameter	Value
Bed void, ϵ_b	0.4
Bed length, L	0.105 m
Number of columns, N_{Column}	8
Number of columns per zone, N_m	2

13.2.2 Process Simulation and Operational Optimization

The decision variables for optimal design and operation of this SMB process include the velocities in the four zones (v_1, v_2, v_3, and v_4) as well as the length of time the process is run each time the inlet/outlet ports are rotated (t_{step}) [126]. The base case values and the bounds of these decision variables are shown in Table 13.2.

Table 13.2: Decision variables for SMB process [126].

Decision Variable	Lower Bound	Base Value	Upper Bound
Zone 1 Velocity, v_1 (m/h)	0.100	6.424	6.424
Zone 2 Velocity, v_2 (m/h)	0.100	4.391	6.424
Zone 3 Velocity, v_3 (m/h)	0.100	4.803	6.424
Zone 4 Velocity, v_4 (m/h)	0.100	3.999	6.424
Rotation time, t_{step} (s)	170	180	200

The axial dimension z was discretized using orthogonal collocation on finite elements with 10 finite elements of 3 collocation points each. The temporal dimension t for each

column was discretized into 5 finite elements of 3 collocation points each. The discretization methods used are identical to those used previously by Kawajiri and Biegler [126] to accurately simulate the physical system. The fully discretized model consists of 13675 equations, and simulation of the resulting large-scale algebraic system without using pseudo-transient models takes 8.9s of CPU time with a peak memory usage of 93.9 MB. The system was solved using the BDNLSOL nonlinear solver in gPROMS [204] in 1455 iterations. While this SMB system could be simulated using an algebraic solver, in general algebraic solvers may fail to find solutions, especially as model complexity and nonlinearity increase.

Simulation of the CSS with the pseudo-transient model takes 6.3s of CPU time with a peak memory usage of 87.0 MB. The system was integrated using the DASOLV solver in gPROMS and required 34 integration steps and 44 interations of the nonlinear solver for reinitialization steps. The solution CPU time is decreased due to an order-of-magnitude reduction in the number of algebraic solver iterations required (the algebraic solver was still employed because DASOLV uses an implicit time integration method). Avoiding the nonlinear solver also allows us to take advantage of the improved convergence properties of pseudo-transient models (see Figure 7.3).

The SMB process was optimized using both the algebraic model and the pseudo-transient formulation. Constraints were included to ensure that the extract purity is at least 97% (not satisfied in the base case) and that at least 80% of the feed is recovered in the extract. Constraints are also included to ensure that the inlet and outlet flowrates are greater than zero. The objective of the optimization calculation is to maximize the velocity of the feed stream, v_F. The optimizations were solved using the sequential quadratic programming (NLPSQP) solver in gPROMS, and the results are reported in Table 13.3.

As expected, optimization with the algebraic model and with the pseudo-transient model converge to the same optimal solution, as the pseudo-transient formulation is solely a

Table 13.3: SMB Process Optimization Results.

	Algebraic	Pseudo Transient
u_1 (m/h)	6.424	6.424
u_2 (m/h)	4.309	4.309
u_3 (m/h)	5.221	5.221
u_4 (m/h)	3.673	3.673
t_{step} (s)	173.7	173.7
u_F (m/h)	0.912	0.912
CPU time (s)	133.1	150.0
NL Iter.	21811	5547
Int. Steps	-	901

mathematical device. The simulation time using the pseudo-transient model was less than that for the algebraic model, but optimization with the pseudo-transient model actually took longer than the same optimization run with the algebraic model, suggesting that the system can be quickly converged between iterations using the algebraic solvers in gPROMS. Though the pseudo-transient approach provides a robust method for solution of the dynamic process and CSS condition, the process model in this case study is assumed isothermal and does not have many nonlinearities. Thus, using the result of each iteration as an initial guess for the subsequent iteration likely is a "good" initial guess and results in a quicker solution time, owing to the use of relatively fast, Newton-type algebraic solvers.

The time relaxation-based optimization algorithm requires a fixed pseudo-time integration horizon between optimization iterations, which slows down optimization with the pseudo-transient system. In principle, pseudo-transient integration steps are only required until the system enters the convergence basin for algebraic solvers, at which point Newton-type, algebraic solvers can be employed for their rapid convergence. Note that Chapter 7 provided a method for such "switching." I expect that a longer pseudo-time integration horizon will be required if the system becomes "more nonlinear," e.g., in the case when the mass transfer coefficients $K_{apps,i}$ in (13.27) increase (causing steeper fronts) and/or when the affinity constants $\mathcal{K}_{i,j}$ in (13.30) increase (which, in turn, affects the bi-Langmuir isotherm).

13.3 Case Study 2: Optimal Design of a Rapid Pressure Swing Adsorption Process

Rapid pressure swing adsorption (RPSA) is a simple process for gas separation, offering relatively high adsorbate productivities for small-scale and low purity separations [183]. Originally proposed for the separation of nitrogen from methane [140], the process comprises two simple steps: (i) pressurization of the bed with feed gas and (ii) countercurrent depressurization through internal purging. The bed is typically composed of small adsorbent particles averaging 200-700 μm in diameter, and the process is typically run with pressurization and depressurization steps of equal durations of about 1-5 seconds.

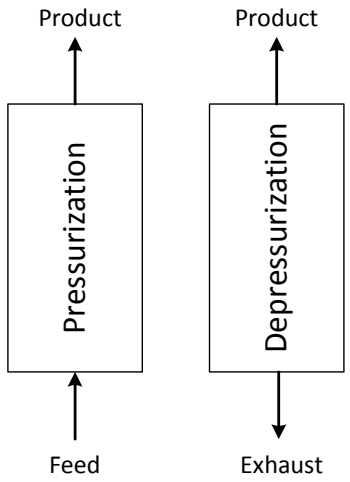

Figure 13.5: Rapid pressure swing adsorption cycle

The effectiveness of RPSA stems from a combination of a fast cycle with a small particle size, leading to steep pressure gradients within the bed. Although these pressure gradients are dynamic, the pressure at the product end is approximately constant, allowing continuous product release. In this case study, I examine an air separation adsorption process using a bed of zeolite 5A, which adrorbs nitrogen, leaving oxygen in the product stream

[5]. I adopt the dynamic model given by Nilchan and Pantelides [183], in which the radial distribution and thermal effects are assumed negligible. Further, the linear driving force model is applied, the fluid is assumed to be an ideal gas, the bed is assumed to be uniform, and the pressure drop in the bed is modeled using Darcy's law. The mass balances in the gas and in the adsorbent are given by:

$$\epsilon_t \frac{\partial C_i(z,t)}{\partial t} = -\frac{\partial (v(z,t)C_i(z,t))}{\partial z} + D_i \frac{\partial^2 C_i(z,t)}{\partial z^2} - \rho_b \frac{\partial q_i(z,t)}{\partial t} \tag{13.54}$$

$$\frac{\partial q_i(z,t)}{\partial t} = k_i(q_i^{eq}(z,t) - q_i(z,t)) \tag{13.55}$$

where v is the superficial gas velocity, C_i is the gas-phase concentration of component i, D_i is the axial dispersion coefficient of component i, ρ_b is the bed density, q_i is the solid-phase concentration of component i, and ϵ_t is the total void bed fraction. Darcy's law is used to approximate the pressure drop through the bed:

$$\frac{\partial P(z,t)}{\partial z} = \frac{-180\mu v(z,t)}{d_p^2} \frac{(1-\epsilon_b)^2}{\epsilon_b^3} \tag{13.56}$$

where d_p is the particle diameter and μ is the gas viscosity. A linear isotherm was used to calculate equilibrium solid-phase concentration:

$$q_i^{eq} = m_i p_i \tag{13.57}$$

$$m_1 = 3.08 \times 10^{-6} \frac{\text{mol}}{\text{kg Pa}} \tag{13.58}$$

$$m_2 = 1.43 \times 10^{-6} \frac{\text{mol}}{\text{kg Pa}} \tag{13.59}$$

where p_i is the partial pressure for component i. The mass transfer coeffient k_i was calculated as:

$$k_i = \frac{15}{r_p^2} \frac{\epsilon_p(1-\epsilon_b)}{\rho_b m_i RT} \frac{\epsilon_p D_p}{\tau_p} \tag{13.60}$$

where r_p is the particle radius and τ_p is the particle tortuosity factor. I assume that the diffusion coeffient can be calculated as [183]:

$$\frac{1}{D_p} = \frac{1}{D_k} + \frac{1}{D_m} \tag{13.61}$$

where D_k is the Knudsen diffusion coefficient (assumed to be constant) and D_m is the molecular diffusion coefficient (assumed to be inversely proportional to pressure). The boundary conditions in the axial dimension depend on the operation stage of the column. During the pressurization step, the boundary conditions are:

$$v(0,t)\left(C_i(0,t) - y_{i,feed}(0,t)\frac{P_{feed}}{RT_{feed}}\right) = D_i\frac{\partial C_i(0,t)}{\partial z} \tag{13.62}$$

$$P(0,t) = P_{feed} \tag{13.63}$$

$$P(L,t) = P_{atm} \tag{13.64}$$

$$\frac{\partial C_i(L,t)}{\partial z} = 0 \tag{13.65}$$

where $y_{i,feed}$ is the mole fraction of component i in the feed, T_{feed} is the temperature of the feed, and P_{feed} is the pressure of the feed. During the depressurization step, the boundary conditions are:

$$\frac{\partial C_i(0,t)}{\partial z} = \frac{\partial C_i(L,t)}{\partial z} = 0 \tag{13.66}$$

$$P(0,t) = P_{exhaust} \tag{13.67}$$

$$P(L,t) = P_{atm} \tag{13.68}$$

where $P_{exhaust}$ is the exhaust pressure. For the process to be at CSS, the conditions in both the gas and solid phases at the end of each full cycle (pressurization and depressurization) must be identical to those at the start. The boundary conditions that must be satisfied in the temporal domain (CSS conditions) are thus:

$$C_i(z,0) = C_i(z,T_{cycle}) \tag{13.69}$$

$$q_i(z,0) = q_i(z,T_{cycle}) \tag{13.70}$$

13.3.1 Pseudo-Transient Formulation

Following the proposed pseudo-transient reformulation strategy again, two new state variables, $\hat{\phi}_C(z,t)$ and $\hat{\phi}_q(z,t)$, are defined such that:

$$\frac{\partial C_{n,i}(z,t)}{\partial t} = \hat{\phi}_C(z,t) \tag{13.71}$$

$$\frac{\partial q_{n,i}(z,t)}{\partial t} = \hat{\phi}_q(z,t) \tag{13.72}$$

The mass balance equations are then reformulated as psuedo-transient equations:

$$\tau_{\phi_C}\frac{\partial \hat{\phi}_C(z,t)}{\partial \hat{t}} = \frac{1}{\epsilon_t}\left(- \frac{\partial (v(z,t)C_i(z,t))}{\partial z} + D_i\frac{\partial^2 C_i(z,t)}{\partial z^2} - \rho_b\frac{\partial q_i(z,t)}{\partial t}\right) - \hat{\phi}_C(x,t) \tag{13.73}$$

$$\tau_{\phi_q}\frac{\partial \hat{\phi}_q(z,t)}{\partial \hat{t}} = k_i(q_i^{eq}(z,t) - q_i(z,t)) - \hat{\phi}_q(x,t) \tag{13.74}$$

$$\hat{\phi}_C(\hat{t}=0) = \hat{\phi}_q(\hat{t}=0) = 0 \tag{13.75}$$

The CSS condition is treated with a dynamic "tearing" procedure:

$$C_{N_{Column}}(x,0) = C_{tear}(x) \tag{13.76}$$

$$q_{N_{Column}}(x,0) = q_{tear}(x) \tag{13.77}$$

$$\tau_{r_c}\frac{\partial C_{tear}(x)}{\partial \hat{t}} = C_{1,i}(x,t_{step}) - C_{tear}(x) \tag{13.78}$$

$$\tau_{r_q}\frac{\partial q_{tear}(x)}{\partial \hat{t}} = q_{1,i}(x,t_{step}) - q_{tear}(x) \tag{13.79}$$

The parameters for the physical system are provided in Table 13.4.

13.3.2 Simulation and Optimization

As given by Nilchan and Pantelides [183], the decision variables for the RPSA process are the particle size, the bed length, the cycle time, and the feed pressure. The base case values and bounds for the decision variables are shown in Table 13.5.

Table 13.4: Parameter values for RPSA process [183].

Parameter	Value
Bed density, ρ_b	800 kg/m^3
Bed void, ϵ_b	0.35
Particle void, ϵ_p	0.55
Tortuosity factor, τ_p	3
Gas viscosity, μ	1.8×10^{-5} N s m^{-2}
Atmospheric pressure, P_{atm}	1 bar
Exhaust pressure, $P_{exhaust}$	1 bar
Feed temperature, T_{feed}	290 K
Feed nitrogen, $y_{1,feed}$	79%
Feed oxygen, $y_{2,feed}$	21%

Table 13.5: Decision variables for RPSA process [183].

Decision Variable	Lower Bound	Base Value	Upper bound
Particle size, d_p (μm)	100	302.5	1000
Bed length, L (m)	0.4	1.0	1.6
Cycle time, T_c (s)	1.0	3.0	5.0
Feed pressure, P_f (bar)	1.0	2.12	3.0

The axial dimension z was discretized using orthogonal collocation on finite elements with 6 finite elements equal to 6 of 3 collocation points each. The model consists of 220 equations, and it was found that the CSS could be approximately reached after simulation of 1,500 cycles. Simulation of 1,500 cycles at the base values of the decision variables required 116.0s of CPU time with a peak memory use of 15.7 MB. 201 iterations of the algebraic solver were used for initialization, and 313,771 integration steps were attempted by the DAE solver.

The time dimension t was then also discretized using orthogonal collocation on finite elements with 7 finite elements equal of 3 collocation points each. The fully discretized model consists of 11,961 equations, and solution of the resulting large-scale algebraic system fails after 114.4s of CPU time (peak memory usage of 86.4 MB) when the algebraic solver (BDNLSOL) reaches a maximum number of iterations. The CSS state was successfully simulated in gPROMS using DASOLV with the pseudo transient model in 16.1s of CPU time with a peak memory usage of 76.6 MB (276 integration steps were attempted by the

DAE solver). The objective function for optimization of the RPSA process is the power used by the process, calculated as:

$$J = \frac{W}{t_{cycle}} \sum_{i=1}^{2} \int_{0}^{t_{cycle}/2} v(0,t) C_i(0,t) dt \tag{13.80}$$

$$W = \left(\frac{k}{k-1}\right) RT_{feed} \left(\left(\frac{P_{feed}}{P_{atm}}\right)^{\frac{k-1}{k}} - 1 \right) \tag{13.81}$$

It was determined that full discretization with 7 finite elements allows for a reasonable balance between simulation time and error in the objective function (compared to the dynamic simulation), shown in Table 13.6, and the same discretization was used for all ensuing optimization routines.

Table 13.6: Objective function error (% deviation from dynamic simulation) in full discretization of RPSA process.

N_{FET}	Power	CPU Time	Eqns	Error
2	106.59	4.0	3811	0.365%
3	106.78	7.2	5441	0.194%
4	106.86	8.6	7071	0.112%
5	106.91	11.5	8701	0.069%
6	106.94	14.8	10331	0.044%
7	106.95	18.0	11961	0.030%
8	106.96	19.1	13591	0.021%
9	106.97	179.6	15221	0.015%
10	106.97	123.2	16851	0.012%
Dynamic	106.98	116.0	220	-

N_{FET} is the number of finite elements of three collocation points each.

The lone constraint during optimization is that the oxygen product stream has a minimum purity of 89%. I first performed a series of optimization runs to determine the optimal value of each decision variable, with the others being held constant at their base case values. A final optimization run was then carried out allowing all four decision variables to be simultaneously optimized. The optimizations were again all carried out using the NLPSQP algorithm in gPROMS. The results of the five optimization runs are reported in Table 13.7.

Table 13.7: RPSA process optimization results.

Opt.	d_p (μm)	L (m)	T_{cycle} (s)	P_{feed} (bar)	Power (W)	Error (%)	CPU time (s)	Opt. Iter.
1	290.29	-	-	-	102.47	0.03	87.9	13
2	-	1.14	-	-	106.65	0.02	23.5	3
3	-	-	5	-	84.81	0.07	17.1	2
4	-	-	-	2.06	96.89	0.03	78.2	4
5	485.14	1.6	5	1.70	52.23	0.04	87.4	9

The optimal value of the objective function found when allowing all four decision variables to change is 52.23 W, considerably smaller than that attained by optimization of any one single decision variable. I also note that CSS simulation using the dynamic model on the optimization results demonstrates that the full discretization method maintains a relatively constant error in the objective function.

13.4 Summary

This chapter presented a novel methodology for modeling, simulation, and optimization of dynamic process models, with a particular focus on periodic processes. The proposed approach involves a two-step reformulation of the dynamic model, consisting of a full discretization of the time and spatial domains, followed by re-casting the resulting system as a pseudo-tansiend model. I provided a theorem to characterize the stability properties of this approach, demonstrating that, under typical assumptions concerning the Lipschitz properties of the original model and the time discretization step, the reformulated system is asymptotically stable in the vicinity of the steady-state solution. The theoretical developments were demonstrated using two extensive case studies, focusing on the simulation and optimization of a simulated moving bed chromatography process and a rapid pressure-swing adsorption system. The case studies showed that my framework can robustly handle the simulation and optimization of large-scale, nonlinear models describing realistic periodic processes.

Chapter 14

Optimization of a Four-Step Pressure Swing Adsorption Process and an Industrial Hydrogen Plant[†]

This chapter examines the modeling and optimization of a four-step pressure swing adsorption (PSA) process for the purification of hydrogen, using the pseudo-transient approach detailed in Chapter 13. The PSA model is particularly relevant to steam-methane reforming (SMR) processes, which produce the bulk of hydrogen in (petro)chemical processing. Building on concepts from Part I, I incorporate the dynamic PSA model into a *multi-resolution* hydrogen plant flowsheet with a detailed SMR model, using pseudo-transient continuation for its reliable simulation and optimization. The combined flowsheet fully accounts for tradeoffs between the reaction (the SMR furnace) and separation (the PSA process) components of the process during optimization. The integration of steady-state and dynamic (cylic) models is non-trivial, and I present a general modeling and optimization framework for process flowsheets containing both classes of models. The presentation in this chapter follows closely the material published in Tsay et al. (2018) [261] and Tsay et al. (2019) [257].

[†]The contents of this chapter are largely based on the following two publications (C.T. is the primary author of both):
C. Tsay, R.C. Pattison, and M. Baldea. A pseudo-transient optimization framework for periodic processes: Pressure swing adsorption and simulated moving bed chromatography. AIChE J., 64(8):2892–2996, 2018.
C. Tsay, A. Kumar, T.F. Edgar and M. Baldea. Integrating steady-state and dynamic models for multi-scale flowsheet optimization: A steam-methane reforming case study. Comput.-Aided Chem. Eng., 47:403–408, 2019.

14.1 Optimal Design and Operation of a Skarstrom-Cycle PSA Process

I consider the optimization of a bench-scale two-bed, four-step pressure swing adsorption (PSA) process for separation of hydrogen from methane, using the model described by Agarwal et al. [2]. This model includes the contributions of thermal effects and pressure losses. A PSA process is designed to cycle between preferentially adsorbing a particular species at high pressure (in this case methane) and regenerating the bed by reducing the total pressure while desorbing the adsorbed species into a separate purge stream. I consider a process with four steps, as shown in Figure 14.1.

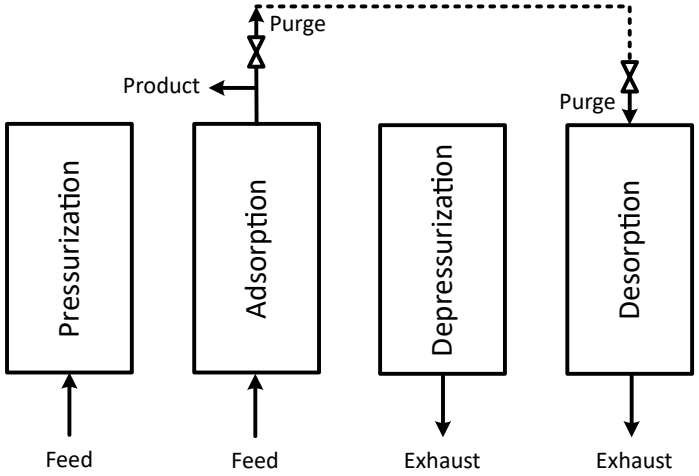

Figure 14.1: Four-step pressure swing adsorption cycle

The four steps of the PSA cycle comprise:

(i) **Pressurization** of the bed with feed gas comprised of 30% hydrogen and 70% methane

(ii) **Adsorption** to produce 99% purity hydrogen product

(iii) **Depressurization** through internal countercurrent purging

(iv) **Desorption** with a low-pressure purge stream

Two beds are operated simultaneously, such that the pressurization and depressurization steps are paired, and the adsorption and desorption steps are paired (the purge stream passes in succession through beds that are undergoing regeneration).

During the pressurization step, high-pressure feed comprised of 30% hydrogen and 70% methane is supplied to the bottom of the bed. Methane is preferentially adsorbed, and hydrogen exits as a product from the top of the bed during the adsorption step. Next, the pressure is reduced to the purge pressure and methane is recovered in the depressurization step. A small portion of the hydrogen product is drawn from the product stream at low pressure to purge additional accumulated methane in the desorption step. I implement the dynamic model given by Agarwal et al. [2], in which radial variations in temperature, pressure, and concentrations are assumed negligible. Further, the linear driving force model is applied, the fluid is assumed to be an ideal gas, the bed is assumed to have uniform thermal and transport properties, and the pressure drop in the bed is modeled using the Ergun equation. Lastly, the gas and solid phases are assumed to be in thermal equilibrium, and the superficial gas velocity is modeled with a linear profile, as proposed by Cruz et al. [58]. The mass balances in the gas and in the adsorbent are given by:

$$\frac{\partial y_i(z,t)}{\partial t} = -v(z)\frac{\partial y_i(z,t)}{\partial z} + D_i\left(\frac{\partial^2 y_i(z,t)}{\partial z^2} - \frac{2}{T(z,t)}\frac{\partial y_i(z,t)}{\partial z}\frac{\partial T(z,t)}{\partial z} + \frac{2}{P(z,t)}\frac{\partial y_i(z,t)}{\partial z}\frac{\partial P(z,t)}{\partial z}\right) - \frac{RT(z,t)}{P(z,t)}\frac{1-\epsilon_b}{\epsilon_b}\rho^b\left(\frac{\partial q_i(z,t)}{\partial t} - y_i(z,t)\sum_{j=1}^{2}\frac{\partial q_j(z,t)}{\partial t}\right) \quad (14.1)$$

$$\frac{\partial q_i(z,t)}{\partial t} = K_i(q_i^{eq}(z,t) - q_i(z,t)) \quad (14.2)$$

$$q_i^{eq}(z,t) = f(y_i(z,t)) \quad (14.3)$$

where v is the fluid velocity, y_i is the gas-phase mole fraction of component i, q_i is the solid-phase concentration, q_i^{eq} is the equilibrium solid-phase concentration, T is the temperature,

P is the pressure, ϵ_b is the bed void fraction, ρ^b is the bed density, and k_i is the lumped mass transfer coefficient for component i. The equilibrium solid-phase concentrations are calculated using the bi-Langmuir isotherm provided by Yang et al. [288] to account for adsorption saturation and interactions between species:

$$q_i = \frac{q_{m1,i} B_{1,i} P_i}{1 + \sum_k (B_1, kP_k)} + \frac{q_{m2,i} B_{2,i} P_i}{1 + \sum_k (B_2, kP_k)} \tag{14.4}$$

$$q_{m1,i} = k_{1,i} \tag{14.5}$$

$$q_{m2,i} = k_{4,i} \tag{14.6}$$

$$B_i = k_{2,i} \exp(k_{3,i}/T) \tag{14.7}$$

$$B_i = k_{5,i} \exp(k_{6,i}/T) \tag{14.8}$$

The parameters for the adsorption of hydrogen and methane on activated carbon are provided in Table 14.1.

Table 14.1: Dual-site Langmuir isotherm parameters for activated carbon [288].

Parameter	Hydrogen	Methane
k_1 (mol/g)	0.01010	0.00576
k_2 (1/atm)	1.90×10^{-5}	7.44×10^{-5}
k_3 (K)	1170.00	2157.02
k_4 (mol/g)	8.73×10^{-4}	7.09×10^{-4}
k_5 (1/atm)	4.98×10^{-6}	2.73×10^{-2}
k_6 (K)	1901.46	1467.70
K_i (1/s)	0.259	0.136
ΔH_i (J/mol)	8420	24124

The energy balance in the gas-phase is given by:

$$(\epsilon_t \rho^g C_p^g + \rho^b C_p^s) \frac{\partial T(z,t)}{\partial t} = -\rho^g C_p^g \epsilon_b v(z) \frac{\partial T(z,t)}{\partial z} + \kappa_L \frac{\partial^2 T(z,t)}{\partial z^2} + \rho^b \sum_{i=1}^{2} (\Delta H_i \frac{\partial q_i(z,t)}{\partial t}) - \frac{2h}{R_b}(T(z,t) - T_w) \tag{14.9}$$

where ϵ_t is the total void fraction, ρ^g is the gas density, ρ_b is the bed density, C_p^g is the vapor heat capacity, C_p^s is the solid heat capacity, κ_L is the thermal diffusivity, ΔH_i is the heat of

adsorption of component i, h is the heat transfer coefficient, R_b is the bed radius, and T_w is the ambient temperature. The Ergun equation was again used to model the pressure drop through the bed (4.39), and is reproduced below with using particle radius (R_p) instead of diameter:

$$\frac{\partial P(z,t)}{\partial z} = \frac{-150\mu}{4R_p^2}\frac{(1-\epsilon_b)^2}{\epsilon_b^3}v(z) + \frac{1.75\rho^g}{2R_p}\frac{1-\epsilon_b}{\epsilon_b^3}v(z)^2 \tag{14.10}$$

The boundary conditions for each operating step are summarized in Table 14.2.

Table 14.2: Boundary conditions for PSA process.

Pressurization	Adsorption	Depressurization	Desorption
$y_i\|_{z=0} = y_{feed,i}$	$y_i\|_{z=0} = y_{feed,i}$	$(\partial\ y_i/\partial z)\|_{z=0} = 0$	$(\partial y_i/\partial z)\|_{z=0} = 0$
$(\partial\ y_i/\partial z)\|_{z=L} = 0$	$(\partial y_i/\partial z)\|_{z=L} = 0$	$(\partial y_i/\partial z)\|_{z=L} = 0$	$y_i\|_{z=L} = y_{purge,i}$
$P\|_{z=0} = P_{high}$	$P\|_{z=0} = P_{high}$	$P\|_{z=0} = P_{low}$	$P\|_{z=0} = P_{low}$
$T\|_{z=0} = T_{feed}$	$T\|_{z=0} = T_{feed}$	$(\partial T\ /\ \partial z)\|_{z=0} = 0$	$(\partial T/\partial z)\|_{z=0} = 0$
$(\partial T/\partial z)\|_{z=L} = 0$	$(\partial T/\partial z)\|_{z=L} = 0$	$(\partial T/\partial z)\|_{z=L} = 0$	$(\partial T/\partial z)\|_{z=L} = 0$
$v\|_{z=0} = v_{feed}$	$v\|_{z=0} = v_{feed}$	$v\|_{z=0} = -v_{reg}$	$v\|_{z=0} = -v_{reg}$
$v\|_{z=L} = 0$	$v\|_{z=L} = 0.125\ v_{feed}$	$v\|_{z=L} = 0$	$v\|_{z=L} =$ -0.6 v_{reg}

For the process to be at cyclic steady state, each step must begin from the conditions at which the previous step ended:

$$\mathbf{X}_{pressurization}(t=0) = \mathbf{X}_{desorption}(t=t_a) \tag{14.11}$$

$$\mathbf{X}_{adsorption}(t=0) = \mathbf{X}_{pressurization}(t=t_p) \tag{14.12}$$

$$\mathbf{X}_{depressurization}(t=0) = \mathbf{X}_{adsorption}(t=t_a) \tag{14.13}$$

$$\mathbf{X}_{desorption}(t=0) = \mathbf{X}_{depressurization}(t=t_p) \tag{14.14}$$

where $\mathbf{X}$ is the spatially distributed vector of states within the adsorption bed (gas-phase concentrations, solid-phase concentrations, and temperature profile).

14.1.1 Pseudo-Transient Formulation

Following the pseudo-transient reformulation strategy presented in Chapter 13, I define three new state variables $\hat{\phi}_y(z,t)$, $\hat{\phi}_q(z,t)$, and $\hat{\phi}_T(z,t)$ such that:

$$\frac{\partial y_i(z,t)}{\partial t} = \hat{\phi}_{y,i}(z,t) \tag{14.15}$$

$$\frac{\partial q_i(z,t)}{\partial t} = \hat{\phi}_{q,i}(z,t) \tag{14.16}$$

$$\frac{\partial T(z,t)}{\partial t} = \hat{\phi}_T(z,t) \tag{14.17}$$

The mass balance equations are then reformulated as pseudo-transient equations:

$$\tau_{\hat{\phi}_y,0}\exp(\gamma t)\frac{\partial \hat{\phi}_{y,i}(z,t)}{\partial \hat{t}} = -v(z)\frac{\partial y_i(z,t)}{\partial z} + D\left(\frac{\partial^2 y_i(z,t)}{\partial z^2} - \frac{2}{T(z,t)}\frac{\partial y_i(z,t)}{\partial z}\frac{\partial T(z,t)}{\partial z} + \frac{2}{P(z,t)}\frac{\partial y_i(z,t)}{\partial z}\frac{\partial P(z,t)}{\partial z}\right) - \frac{RT(z,t)}{P(z,t)}\frac{1-\epsilon_b}{\epsilon_b}\rho^b\left(\frac{\partial q_i(z,t)}{\partial t} - y_i(z,t)\sum_{j=1}^{2}\frac{\partial q_j(z,t)}{\partial t}\right) - \hat{\phi}_{y,i}(z,t) \tag{14.18}$$

$$\tau_{\hat{\phi}_q,0}\exp(\gamma t)\frac{\partial \hat{\phi}_{q,i}(z,t)}{\partial \hat{t}} = k_i(q_i^{eq}(z,t) - q_i(z,t)) - \hat{\phi}_{q,i}(z,t) \tag{14.19}$$

$$\tau_{\hat{\phi}_T,0}\exp(\gamma t)\frac{\partial \hat{\phi}_T(z,t)}{\partial \hat{t}} = \frac{-1}{\epsilon_t\rho^g C_p^g + \rho^b C_p^s}\rho^g C_p^g \epsilon_b v(z)\frac{\partial T(z,t)}{\partial z} + \kappa_L\frac{\partial^2 T(z,t)}{\partial z^2} + \rho^b\sum_{i=1}^{2}(\Delta H_i\frac{\partial q_i(z,t)}{\partial t}) - \frac{2h}{R_b}(T(z,t) - T_w) - \hat{\phi}_T \tag{14.20}$$

where γ is a constant introduced as in (13.25) to improve the system stability. Figure 14.2 shows the pseudo-time dynamics of the reformulated system for various values of γ ($\gamma = 5$, 10, and 20) at $z = L$ of the pressurization step.

The effects of altering γ are most prevalent in the initialization of the pseudo-transient system before the CSS condition is fully enforced (before $\hat{t} = 10^4$ in Figure 14.2). As expected, increasing γ allows the time-discretized points to be solved sequentially by separating

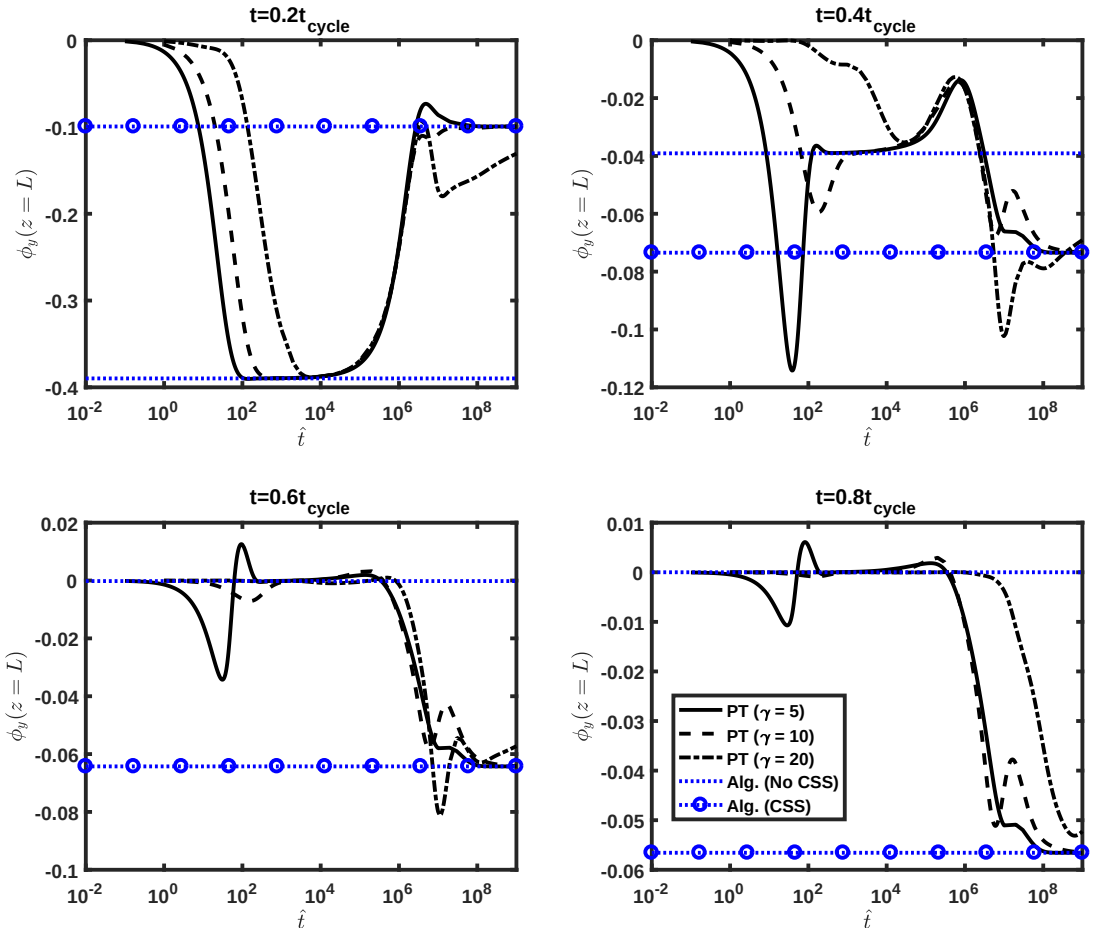

Figure 14.2: Effect of different values of γ on the pseudo-time stability in solving the pseudo-transient system. PT refers to the pseudo-transient model and Alg. refers to the original algebraic system. The system first is at first near the solution of the algebraic system with no CSS condition (as the CSS condition is initially not enforced by the dynamic tearing procedure (13.23)–(13.24)), then gradually transitions to the solution of the algebraic system with CSS condition as the dynamic tear reaches its steady state at the end of the plots.

their dynamics, evidenced by the decrease in fluctuations resulting from nearing instability of the pseudo-transient dynamics (as shown in Theorem 13.1.1). Simulation with a value for γ much lower than $\gamma = 5$ results in an unstable pseudo-transient system, with the state variables $\hat{\phi}$ increasing/decreasing exponentially past their bounds rather than exhibiting an

oscillatory but stable behavior that ultimately results in them reaching a steady-state value. However, increasing γ also slows down the dynamics of the overall pseudo-transient system, which in turn increases computational cost by lengthening the pseudo-time integration horizon required to reach steady state. It can be seen in Figure 14.2 that the model with $\gamma = 20$ has very slow dynamics, the time scale of the pseudo-transient dynamics now coincides with that of the (pseudo-transient) dynamic tearing equations described below, resulting in an apparent high-order response that reflects the competing effects of the pseudo-transient dynamics and of the "torn" CSS constraints.

The CSS condition is again treated with a dynamic "tear," such that the boundary conditions of the time domain become:

$$\mathbf{X}_{pressurization}(t=0) = \mathbf{X}_{tear} \tag{14.21}$$

$$\mathbf{X}_{adsorption}(t=0) = \mathbf{X}_{pressurization}(t=t_p) \tag{14.22}$$

$$\mathbf{X}_{depressurization}(t=0) = \mathbf{X}_{adsorption}(t=t_a) \tag{14.23}$$

$$\mathbf{X}_{desorption}(t=0) = \mathbf{X}_{depressurization}(t=t_p) \tag{14.24}$$

$$\tau_r \frac{\partial \mathbf{X}_{tear}}{\partial \hat{t}} = \mathbf{X}_{desorption}(t=t_a) - \mathbf{X}_{tear} \tag{14.25}$$

In addition, the composition of the purge stream leaving the bed in the adsorption step must be the same as the purge stream entering the bed in the desorption step, which is also handled using a dynamic tear. The boundary condition is thus formulated:

$$y_{desorption,i}(z=L) = y_{purge,i} \tag{14.26}$$

$$\tau_{r,2} \frac{\partial y_{purge,i}}{\partial \hat{t}} = y_{adsorption,i}(z=L) - y_{purge,i} \tag{14.27}$$

The parameters for the physical system are provided in Table 14.3

Table 14.3: Parameter values for PSA process [2].

Parameter	Value
Gas Viscosity, μ	3.73×10^{-8} kg/m/s
Bed Porosity, ϵ_b	0.404
Bed Density, ρ_b	426.7 kg/m^3
Bed Length, L	1 m
Bed Radius, R_b	0.25 m
Solid Heat Capacity, C_{ps}	1046.7 J/kg/K
Particle Porosity, ϵ_p	0.546
Particle Density, ρ_p	716.3 kg/m^3
Particle Radius, R_p	5.41 mm
Thermal Diffusivity, K_L	1.2×10^{-6} J/m/s/K
Diffusivity, D_i	1.3×10^{-5} m^2/s
Heat Transfer Coefficient, h	60 J/m^2/s/K
Wall Temperature, T_w	300 K

14.1.2 Simulation and Optimization

As given by Agarwal et al. [2], the decision variables for the PSA process are the high operating bed pressure, low operating bed pressure, pressurization step time, adsorption step time, feed velocity, and regeneration velocity. The base case values and enforced bounds for the decision variables are shown in Table 14.4.

Table 14.4: Decision variables for PSA process.

Decision Variable	Lower Bound	Base Value	Upper Bound
High operating pressure, P_H (Pa)	4.5×10^5	6.0×10^5	7.5×10^5
Low operating pressure, P_L (Pa)	1.0×10^5	1.5×10^5	2.0×10^5
Pressurization step time, t_p (s)	2	5	8
Adsorption step time, t_a (s)	40	50	60
Feed velocity, v_{feed} (m/s)	0.05	0.10	0.15
Renegeneration velocity, v_{reg} (m/s)	0.04	0.05	0.06
H_2 Recovery	-	8.3%	-
H_2 Purity	-	97.2%	-

The time dimension t was discretized using orthogonal collocation on finite elements with 30 finite elements of 3 collocation points each, and the axial dimension z was discretized using finite differences in the reverse direction of flow for each PSA step. Solution of the fully discretized (algebraic) system was attemped using the algebraic solver in gPROMS [204], and

it was found that convergence depends on the initial guess provided for the variable types. Assuming that uniform initial guesses are used for each variable type (as is the default in gPROMS), Figure 14.3 illustrates the convergence of the algebraic model from various initial guesses of temperature and pressure, with the initial guesses for the other variables fixed; the initial guesses for all mole fractions were set to 0.5, and the initial guesses for concentrations in the solid phase were set to 1 mol/kg.

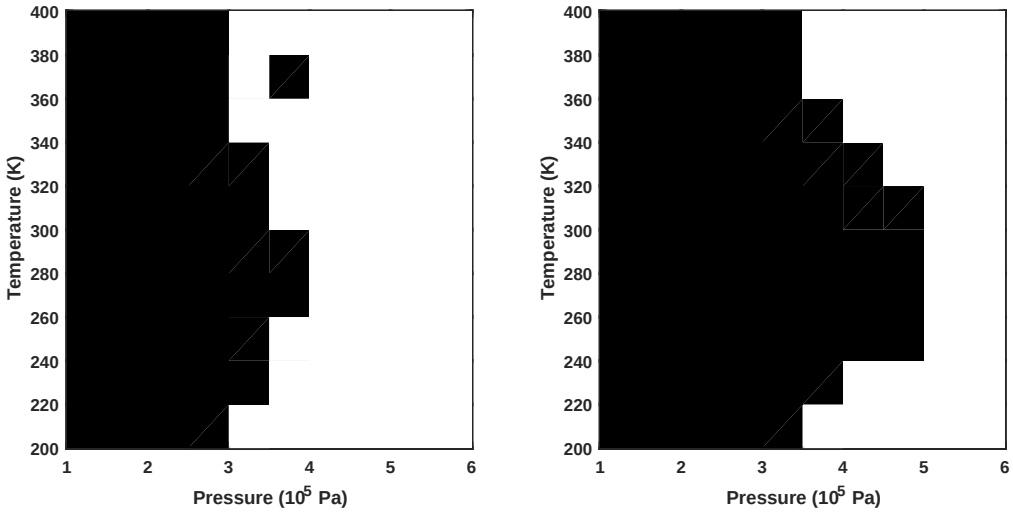

Figure 14.3: Convergence basins for using a nonlinear algebraic solver to simulate the fully discretized PSA process. The plot on the left corresponds to using 12 discretization points for the temporal and axial dimensions, and the plot on the right corresponds to 24 discretization points. The white areas represent initial guesses that converge to the solution.

The convergence basins (Figure 14.3) also show the increase in system complexity—and corresponding decrease in convergence basin size—arising with an increase in the number of discretization points used for the axial and temporal dimensions, which may be important for model accuracy. For instance, a fine axial discretization may be required if the mass transfer coefficients K_i in (14.2) are large (or the diffusion coefficients D_i in (14.1) are small), introducing steep fronts in the solution. In this case study, calculation of the value of hydrogen recovery (the objective function for optimization) was found to be highly dependent

on the number of discretization points in the axial dimension, as seen from simulations using the pseudo-transient model presented in Table 14.5.

Table 14.5: Fully discretized PSA process simulation using pseudo-transient model for various levels of axial discretization (30 discretization points used for temporal domain).

Axial Disc.	H_2 Recovery	CPU Time	Eqns	Δ
5	6.49%	12.0	7781	-
10	7.50%	32.9	14246	15.54%
15	7.90%	79.6	20711	5.28%
20	8.11%	170.5	27176	2.65%
25	8.24%	208.5	33641	1.56%
30	8.31%	324.6	40106	0.93%
35	8.37%	408.1	46571	0.69%
40	8.41%	724.5	53036	0.49%

Going forward, I use 30 discretization points for the axial dimension, as the change in computed H_2 recovery from 25 to 30 discretization points was less than 1%. This fully discretized model consists of 40106 equations. Simulation of the CSS state with the pseudo-transient model requires 324.6s of CPU time with a peak memory usage of 449.0 MB. The system was integrated using the DASOLV algorithm in gPROMS [204] and required 600 integration steps and 2344 interations of the nonlinear solver.

The pseudo-transient model was then optimized using the aforementioned time relaxation based algorithm. The objective for optimization was to maximize hydrogen recovery, defined as the moles of hydrogen produced in the product stream (adsorption step) divided by the number of moles entering the bed in the pressurization and adsorption steps. The lone constraint during optimization is that the hydrogen product stream have a minimum purity of 99% (not satisfied in the base case design). The system was optimized using the NLPSQP algorithm in gPROMS, and the optimization calculation required 1,363s of CPU time, in 5 iterations with a peak memory usage of 736.9 MB. I note that while the CPU time required for optimization here is significantly longer than the 184.5s reported by Agrawal et al. [2], a reduced-order model was used in the work by Agrawal et al., resulting in loss

of accuracy as the decision variables moved away from their initial guesses [2] (which were used to generate the reduced-order models). I also note that a bi-Langmuir isotherm is used in this study, which increases the system complexity, and the constraint for hydrogen purity was set at 99%. The optimization results are reported in Table 14.6.

Table 14.6: PSA process optimization results.

Decision Variable	Lower Bound	Optimal Value	Upper Bound
High operating pressure, P_H (Pa)	4.5×10^5	7.5×10^5	7.5×10^5
Low operating pressure, P_L (Pa)	1.0×10^5	1.0×10^5	2.0×10^5
Pressurization step time, t_p (s)	2	2	8
Adsorption step time, t_a (s)	40	60	60
Feed velocity, v_{feed} (m/s)	0.05	0.077	0.15
Renegeneration velocity, v_{reg} (m/s)	0.04	0.042	0.06
H_2 Recovery	-	13.8%	-
H_2 Purity	-	99.0%	-

While previous attempts at fully discretized optimization of similar four-step PSA processes rely on surrogate or reduced-order models [2, 109], the proposed pseudo-transient approach allows optimization of the full-order nonlinear, distributed, and highly coupled PSA model. Furthermore, the computations provide exact gradient matrices for optimization and are completed relatively quickly (in comparison to sequential optimization methods with full-order models [120, 287]), and the models can be easily coupled with steady-state pseudo-transient unit models for integrated flowsheet optimization. The decision variable bounds used for optimization were wider than those used in previous work by Agrawal et al. [2], as that work used tighter bounds to maintain accuracy of the reduced-order models. In the optimal design (Table 14.6), the high operating pressure and adsorption step times are at their upper bounds, and the low operating pressure and pressurization step times are at their lower bounds, thus maximizing the ratio between the high and low pressures. The optimization process increased the H_2 product purity to the required 99%, and the objective function, H_2 recovery, was increased by 66% from the base case. The mole fractions of

methane along the PSA column for all four steps are shown for the base case design in Figure 14.4 and the optimized process design in Figure 14.5.

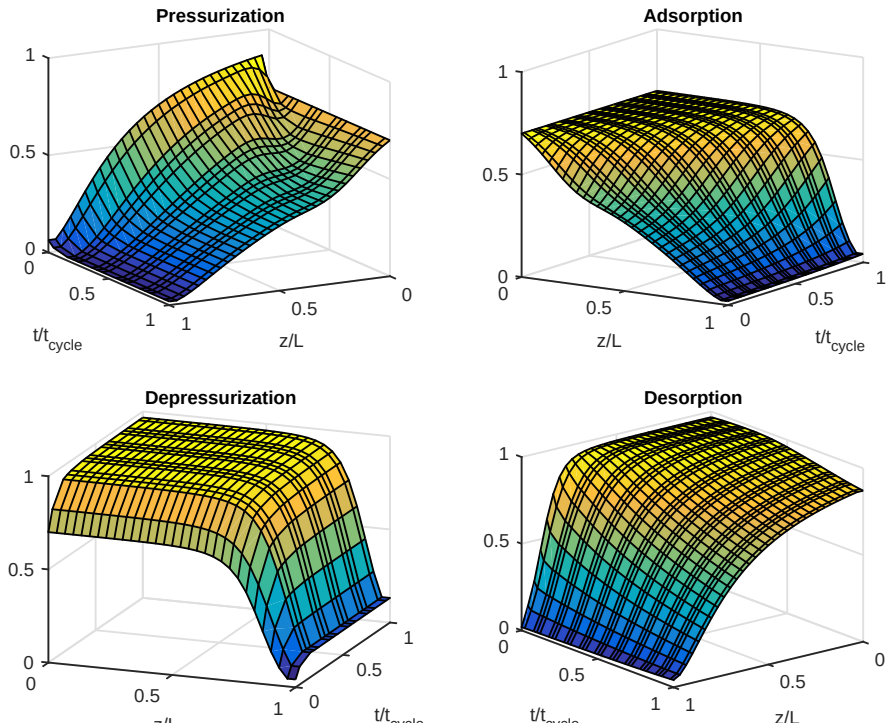

Figure 14.4: Methane mole fractions along the PSA column for the base case design.

14.2 Integrating Steady-State and Dynamic Models for Optimal Design and Operation of A Steam-Methane Reforming Process

Hydrogen is consumed in large quantities in many chemical and petrochemical processes, such as ammonia production and crude oil processing. About 80% of the hydrogen used for these applications is produced through steam reforming of natural gas [238], which produces a mixture of hydrogen and carbon monoxide known as "syngas". The process occurs in large furnaces called steam-methane reformers (SMRs). High-purity hydrogen is

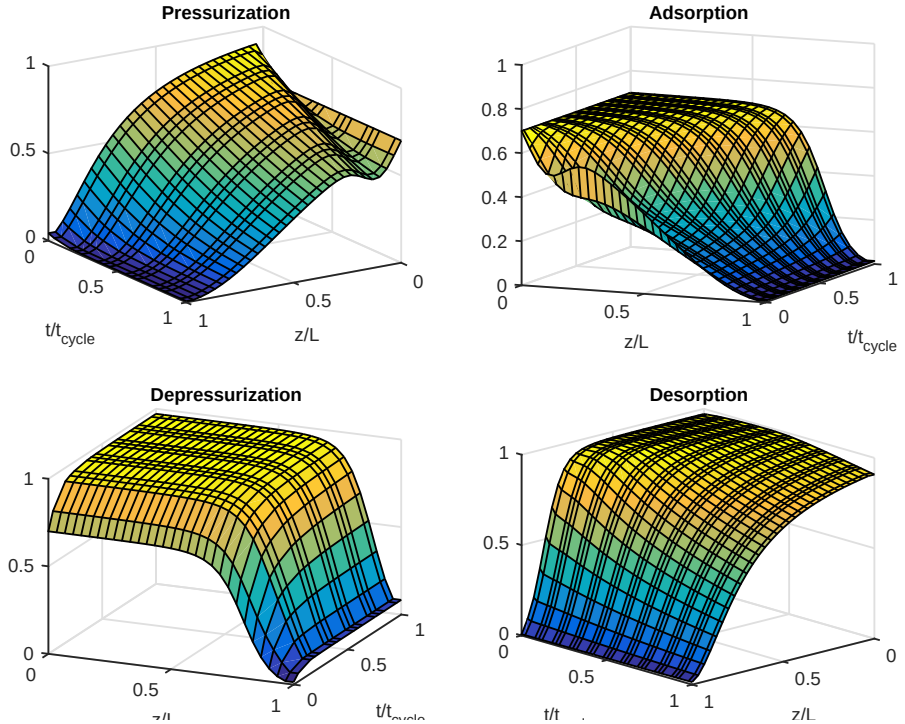

Figure 14.5: Methane mole fractions along the PSA column for the design after optimization.

typically separated from syngas using a PSA process. The combined process has been widely adopted due to its high theoretical energy efficiency. However, in practice many hydrogen-production plants operate inefficiently due to non-ideal behavior in key process units. This inefficiency can translate to large energy costs: a typical SMR process produces 100 million standard cubic feet of hydrogen and consumes ∼105 GJ of energy daily [143]. As a consequence, it is of utmost importance that such plants be designed and operated to maximize process efficiency.

In an effort to minimize the deviation between the theoretical energy efficiency predicted during process design and the realized efficiency, Kumar et al. [143] adopted the *multi-resolution* approach, as described in Chapter 4. Following this framework, a physics-based model of the SMR was used to carry out the optimization of the full process flowsheet,

accounting for non-idealities in the form of a non-uniform temperature distribution in the furnace. However, the key PSA separation step was modeled assuming a constant, known recovery and purity of hydrogen product. Subsequently, a sensitivity analysis revealed that the overall thermal efficiency of the plant was still highly dependent on the value assumed for PSA hydrogen recovery. This finding is also supported by Agarwal et al. [2], who suggest that a detailed PSA model should be included into flowsheet optimization to capture trade-offs between the PSA separation performance and other parts of the process. There are an increasing number of cyclically operated, dynamically intensified processes [17], but cyclic dynamics have not been integrated at the flowsheet level due to challenges in converging their mathematical models.

Motivated by the above, I now formulate a multi-resolution model of an industrial hydrogen plant including both a detailed, physics-based SMR furnace and the (fully discretized) dynamic PSA model described above. The SMR furnace is discretized in three spatial dimensions, while the PSA model is discretized temporally, and in a single axial dimension. This case study demonstrates that these large-scale models can be robustly simulated and optimized in conjunction as a pseudo-transient flowsheet model, and the resulting optimization procedure simultaneously identifies the optimal operating profiles of the SMR and PSA. The inclusion of these details enables the variable performance of these key process units to be considered during the identification of optimal design specifications at the flowsheet level.

Process Flowsheet Description. The process flowsheet for a reforming-based hydrogen production plant is shown in Figure 1.

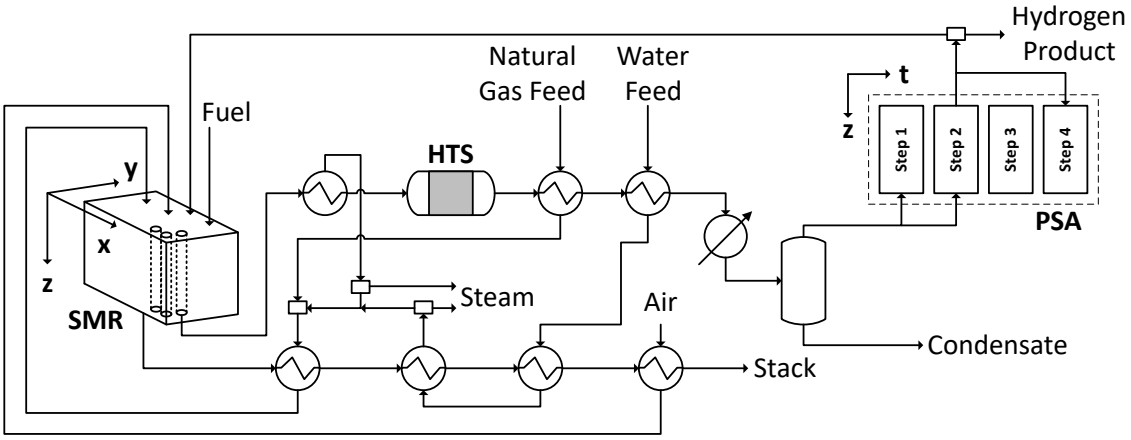

Figure 14.6: Diagram of a reforming-based hydrogen production plant.

Natural gas (assumed to be methane) serves as both the fuel and process feedstock entering the SMR furnace. Within the tubes of the furnace, three chemical reactions occur:

$$\mathrm{CH_4 + 2H_2O \leftrightarrow 4H_2 + CO_2}; \quad \Delta H_{298K} = 165 \frac{\mathrm{kJ}}{\mathrm{mol}} \tag{14.28}$$

$$\mathrm{CH_4 + H_2O \leftrightarrow 3H_2 + CO}; \quad \Delta H_{298K} = 206 \frac{\mathrm{kJ}}{\mathrm{mol}} \tag{14.29}$$

$$\mathrm{CO + 2H_2O \leftrightarrow H_2 + CO_2}; \quad \Delta H_{298K} = -41.2 \frac{\mathrm{kJ}}{\mathrm{mol}} \tag{14.30}$$

where (14.28) and (14.29) correspond to the endothermic reforming reactions and (14.30) corresponds to the exothermic water-gas shift reaction.

On the tube side of the SMR, the hot syngas product exits the SMR and passes through a heat exchanger to recover heat by producing steam. The stream then passes through a high-temperature shift reactor (HTS) to further increase the hydrogen yield, before passing through two additional heat exchangers to preheat the process feed natural gas and water. A steam separator (SS) then removes the water (condensate) from the syngas stream, with the rest of the stream sent to the PSA unit for separation of high-purity hydrogen product. Some of the product hydrogen is recycled and mixed with the process feed, while the rest is compressed for storage or distribution.

On the fuel side of the SMR, combusted fuel exits the furnace and passes through multiple heat exchangers to recover heat. This heat integration accomplishes two tasks: pre-heating the air required for combustion in the furnace and producing additional superheated steam. A portion of the steam is sent to the SMR, while the rest can be exported from the process. The mixed process feed comprising process steam, feed natural gas, and hydrogen recycle from the PSA is fed into the SMR tubes. The cooled flue gas is discharged to a flue stack and vented to the atmosphere.

14.2.1 Steam-Methane Reforming Furnace Model

I adopt the physics-based furnace model proposed by Latham et al. [147] and extended by Kumar et al. [142] was used in this work, which is illustrated in Figure 14.7. In this model, the furnace is divided into groups of six tubes, with a flue gas section on either side. The flue gas side comprises the region of the furnace outside of the tube (consisting of burnt fuel) and the furnace refractory wall. In total, there are 21 groups of tubes (7 x 3) and 24 flue gas sections (8 x 3). Each tube and flue gas section is modeled using a plug-flow assumption with 15 elements. As the effect of combustion is stronger near the top of the furnace, temperatures and gas composition change sharply near the top of the furnace. A non-uniform spatial discretization is thus employed, with the top ten sections having a height of 0.625 m each and the bottom five a height of 1.25 m each. Latham et al. (2011) found that no significant improvement in the model resolution is obtained by increasing the number of elements beyond 15.

Furnace balancing is performed with respect to the temperatures of these tube groups by manipulating the fuel and the air flow rates to the corresponding burner groups. This simplification lowers the model complexity and can be justified by the observation that the tube wall temperatures of adjacent tubes tend to have similar values and dynamics [142]. A

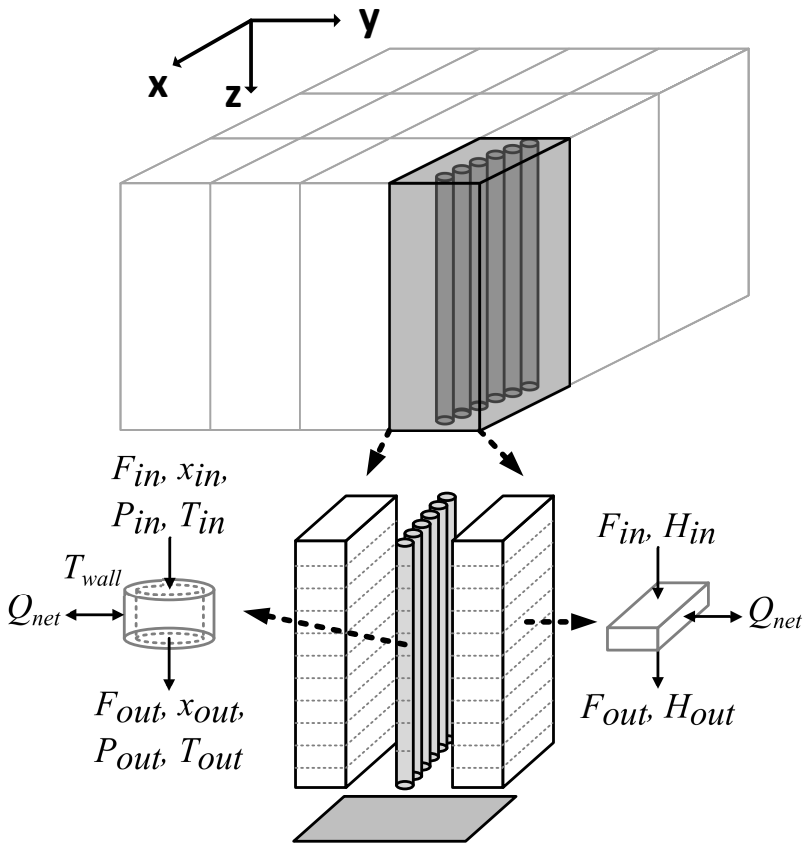

Figure 14.7: The SMR furnace is divided into flue gas and tube sections, which are then each discretized into 15 vertical plug-flow elements. For clarity, the actual number of SMR sections is nto depicted. Likewise, the furnace refractory wall is not pictured.

tube group temperature is defined as the average of the maximum tube wall temperatures of the tubes in that group. Although the tubes within a tube group are lumped together for modeling purposes, radiation for all 366 tubes is still individually modeled.

Modeling of Flue Gas Sections. The fuel mixture entering each burner group is assumed to under go complete and isothermal combustion, and the energy released from combustion in each flue gas section is non-uniformly distributed among the vertical sections. Heat is also lost from each section due to radiation and convection, and the total heat can be computed as $Q_{net} = Q_{combust} - Q_{rad} - Q_{conv}$, where Q_{rad} is the net heat lost by radiation and

Q_{conv} is the net heat lost by convection. The total heat Q_{net} is treated with the continuation parameter $\hat{\alpha}$ as described by (3.11)–(3.3.2). Each flue gas section is modeled as a well-mixed volume, and the temperature is treated as a pseudo-transient state variable with dynamics given by the residual of the energy balance:

$$\tau_e \zeta \frac{dT}{d\hat{t}} = F_{combust} \sum_j x_j \int_T^{T_{in}} C_{p,j} dT' + \hat{\alpha} Q_{net} \tag{14.31}$$

where $F_{combust}$ is the molar flow rate after combustion, and ζ is a unit-valued scalar ensuring consistence of units between the right-hand side and the left-hand side. Radiation is modeled using the Hottel zone method, which employs pairwise parameters describing the radiative flux between each pair of sections. Model parameters are fitted to operating data from an industrial furnace [142]. No reactions (other than combustion) occur in the flue gas sections, and the species mole balance is trivial for the plug-flow sections.

Modeling of Tube Sections. On the tube side, the mass balance accounts for the chemical reactions in (14.28)–(14.30). Each tube section is assumed to be a well-mixed reactor, and the mass and energy balances for each section are expressed as:

$$0 \;=\; F_{j,in} - F_{j,out} + \hat{\alpha}\big(\pi r_{in}^2 \Delta z \rho^c \eta \sum_k \gamma_{j,k} r_k\big) \tag{14.32}$$

$$\tau_e \zeta \frac{dT}{d\hat{t}} = \sum_j F_{j,in} \int_T^{T_{in}} C_{p,j} dT' + \hat{\alpha}\big(h_{tg} 2\pi r_{in} \Delta z (T_{in,wall} - T) - \pi r_{in}^2 \Delta z \rho^c \eta \sum_k \gamma_{j,k} r_k \Delta H_k\big) \tag{14.33}$$

where j indexes the component species and k the reactions (14.28)–(14.30). $\gamma_{j,k}$ is the stoichiometric coefficient of the j^{th} species in the k^{th} reaction, and η is an effectiveness factor assumed to be 0.1 to account for catalyst diffusion resistance. Δz is the height of the tube section and r_{in} is its inner radius.r_k denotes the rate of reaction k, h_{tg} the convective heat transfer coefficient, ρ^c the catalyst density, and $T_{in,wall}$ the temperature of the inner side

of the tube wall. The Ergun equation (4.39) is again used to model pressure losses. The Kirchoff equation is used to predict the effect of temperature on reaction enthalpy:

$$\Delta H_k(T) = \Delta H_k(T = 298\text{K}) + \sum_j \int_{298}^{T} \gamma_{j,k} C_{p,j} dT' \tag{14.34}$$

Enthalpy balances describe the temperatures at the outside of the tube wall, at the inside of the tube wall, and inside the tube. The heat input to the outside of the tube wall comprises the radiation and convection from the other sections of the furnace, $Q_{net} = Q_{rad} + Q_{conv}$. Further details of the model equations can be found in [143] and [142].

14.2.2 Pressure Swing Adsorption Model

The pressure-swing adsorption model described in Section 14.1 is adopted to purify the hydrogen product, with slight modification for the larger application. Two beds are operated in a synchronized manner, such that the pressurization/depressurization steps occur simultaneously, and the adsorption/desorption steps are paired (a small portion of the product stream is used to purge the bed undergoing regeneration). Therefore, one of the paired PSA beds receives the feed stream at all times during the cycle.

The feed to the PSA unit is assumed to be composed of hydrogen, methane, carbon dioxide, and carbon monoxide. The bulk of the water is removed by the steam separator (Figure 13.4), and the remainder is typically removed by a layer of alumina or silica before it reaches the PSA bed(s). The bottom half (closer to the feed) of the bed is packed with activated carbon, while the rest of the bed is packed with Zeolite 5A. The activated carbon layer removes carbon dioxide and methane, while the zeolite layer removes carbon monoxide. Notably, carbon dioxide cannot be readily desorbed from the zeolite layer and should be adsorbed only by the activated carbon. A linear driving force model is again applied, and

the multi-site Langmuir model is used to represent adsorption equilibrium:

$$\frac{q_i}{q_{i,max}} = a_i K_i P \left[1 - \sum_i \left(\frac{q_i}{q_{i,max}}\right)\right]^{a_i} \tag{14.35}$$

where the parameters for the four components and two adsorbents are given by Ribeiro et al. [209]. I also follow the model reduction strategy of [209], with several key simplifications that are summarized below. Only axial variations are considered, with radial variations in temperature, pressure, and concentrations assumed to be negligible. Micropore diffusion is assumed to be the main contributor to mass transfer resistance, and film mass transfer and macropore diffusion are considered negligible. With this assumption, there are only two phases modeled: the gas phase and the adsorbed phase. The heat capacity and viscosity of the gas are assumed to be constant throughout the bed.

Thermal equilibrium is assumed, such that the gas and solid phases have the same temperature. The model is thus reduced to only having one energy balance equation. Each section (activated carbon and zeolite) of the bed is assumed to have constant and uniform thermal and physical properties, and the pressure drop through the bed is modeled using the Ergun equation. For more details on these model assumptions and an analysis of the introduced error, the reader is referred to [209].

14.2.3 Other Process Units

Following the paradigm of multi-resolution flowsheet modeling (Chapter 4), the remaining unit operations are less crucial to the process thermal efficiency, and simpler mathematical models are employed. The pseudo-transient process models for the remainder of the process flowsheet described by Kumar et al. [143] are used, with assumptions summarized here. The HTS reactor is assumed to be adiabatic and is modeled as a plug-flow reactor with a commercial ferrochrome catalyst. The steam separator is modeled as an isothermal

flash that is cooled close to ambient temperature. The multi-stage compression of the hydrogen product is modeled as a series of compressors with inter-coolers. Heat exchangers are modeled using a log-mean temperature difference. Standard pseudo-transient models for compressors and heat exchangers [190] are employed.

14.2.4 Flowsheet Optimization

At the process level, there are multiple recycle streams for heat/material integration that are decoupled in pseudo-time using dynamic tear equations (Eqns. (12)-(13)). The connection between the steady-state and periodic process units is mathematically modeled using a time average of the inputs and outputs of the periodic units:

$$\mathbf{X}_{steady\text{-}state} = \int_0^{t_{cycle}} \mathbf{X}(t)dt \tag{14.36}$$

In essence, (14.36) enforces the principle of dynamic process intensification, where the cyclic steady state corresponds to desired production and product quality *on the average*. In practice, this connection requires storage capacity and a control strategy that can buffer the distribution of the inputs to/outputs of the periodic PSA process.

The process is assumed to operate with ten PSA beds (five pairs). Each PSA bed is fed at a constant rate for $t_{cycle}/2$ (pressurization and adsorption steps), and thus the flow rate to each PSA bed during the pressurization and adsorption steps is one-fifth the vapor flow from the steam separator. The objective function for optimization is the thermal efficiency of the plant, computed as:

$$\eta_{H_2} = \frac{E_{H_2}}{E_{NG} - Q_{steam} + W_{comp}} \tag{14.37}$$

where E_{H_2} and E_{NG} respectively denote the lower heating values of the hydrogen product and the total natural gas fed to the system. Q_{steam} is the thermal value of the excess steam generated for export, and W_{comp} represents the energy required by the compressors. A higher

operating temperature generally increases the thermal efficiency; however, the maximum temperature is constrained by the material limits of the furnace tubes. The furnace is balanced before flowsheet optimization by identifying the fuel flow rates to the 24 flue gas sections that minimize the spread of the distribution of tube temperatures.

In total, seven constraints are enforced for flowsheet optimization:

(i) The hydrogen production rate must be at least 3700 kmol/h (74.2 MMSCFD)

(ii) The maximum tube temperature in the furnace is at most 1200 K

(iii) The feed temperature to the SMR is between 725 and 900 K

(iv) The steam-to-carbon ratio in the SMR feed is between 2.0 and 6.0

(v) The feed temperature to the HTS reactor is between 570 and 730 K

(vi) The ratio of water to hydrogen molecules in the HTS reactor is at least 0.3

(vii) The PSA product is composed of at least 99% hydrogen

These constraints ensure that the equipment operates within the ranges found in industrial units, that carbon formation on the reformer catalyst is avoided, and that no methanation occurs in the HTS reactor. I note that Kumar et al. [143] constrained the hydrogen production rate to be at least 4058 kmol/h; however, a lower constraint value is used in this work, since the PSA process of interest exhibits significantly lower recovery. Using the same PSA adsorption parameters, Ribeiro et al. [209] found hydrogen recoveries of 50-60% for a $\sim$99.9% hydrogen product.

The decision variables at the flowsheet level comprise the flow rates of natural gas fuel, process feed, process steam, feed water, and the split ratio of preheated feed water into

the steam generators. For the PSA unit, the decision variables include the lengths of the pressurization and adsorption steps, the high and low operating pressures, the size of the packed bed, and the purge flowrate.

14.2.5 Results and Discussion

The full model was implemented in gPROMS version 5.1.4 [204]. The axial domain of the PSA bed is discretized into 20 points, using finite differences in the reverse direction of gas flow. The physical time domain is discretized into 30 finite elements of three collocation points each. In total, the flowsheet contains approximately 66,000 equations and 14 decision variables. The optimization procedure used over 3.5 GB RAM at its peak, highlighting the importance of reduced-space optimization (where only sensitivities of the objective function and inequality constraints to decision variables are stored). The process performance at the optimal point is reported in Table 14.7. Case 1 and Case 2 denote previous results in the literature, for which the PSA recovery and purity were assumed to be known a priori and fixed during flowsheet optimization.

Table 14.7: Comparison of SMR process with previous work [143].

	Case 1	Case 2	This Work
PSA H_2 Purity	99.95%	99.95%	99.1%
PSA H_2 Recovery	90%	80%	58%
Thermal Efficiency (η_{H_2})	91.23%	88.37%	78.97%

The PSA recovery and purity found in this work are comparable to those given by Ribeiro et al. [209], but significantly lower than the values assumed by Kumar et al. [143] during process flowsheet optimization. The purity and recovery were found to be strongly dependent on the properties of the inlet stream to the PSA unit, which changes during optimization iterations, emphasizing the importance of using predictive models during flowsheet optimization. Even with the lower hydrogen recovery, the thermal efficiency of the

process remains near 80%, as the process flowsheet was optimized simultaneously, finding the optimal operating conditions for the rest of the units with the predicted PSA performance.

Integrating the PSA model into the multi-resolution process flowsheet allows the separation section to be designed simultaneously with the rest of the process. The detailed model also provides insight into the behavior of the process unit at its cyclic steady state, aiding in the understanding of the process for potential troubleshooting procedures. Furthermore, the optimization results demonstrate that important process units can be designed simultaneously with making flowsheet-level decisions, providing a more realistic view of the phenomena occurring in the process and accounting for interdependencies among the performances of various unit operations. Figure (14.8) shows the hydrogen concentration throughout the four steps of the modeled PSA process at the optimal point. I emphasize that these profiles were obtained simultaneously with the full process flowsheet optimization procedure.

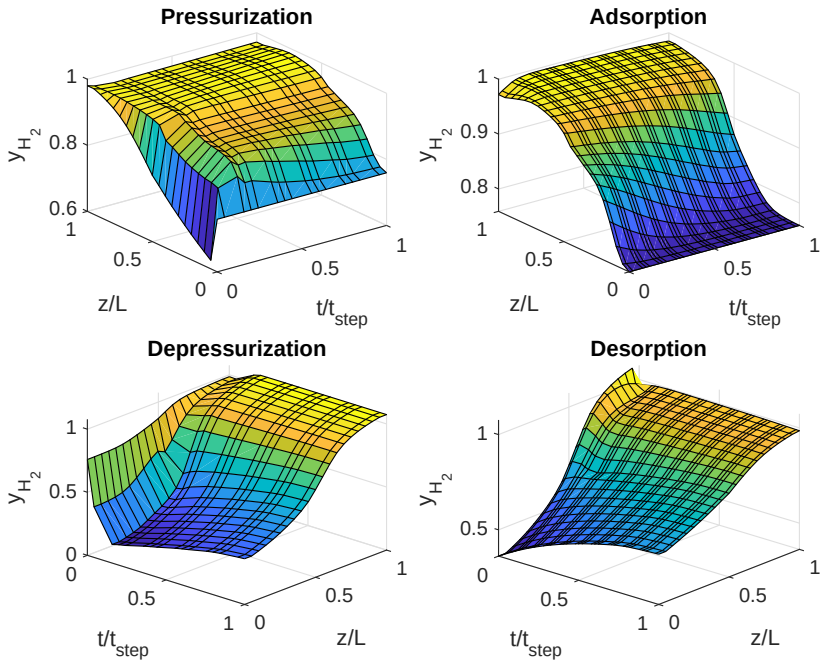

Figure 14.8: Hydrogen concentration profiles through the PSA cycle.

14.3 Summary

In this chapter, I described the modeling and optimization of a four-step pressure swing adsorption process (PSA), as well as a relevant application in hydrogen production. Simulation of the process model provided insight into the dynamic stability of the pseudo-transient reformulation, as analyzed in Chapter 13. I showed how the time constants can be set as a decaying function of the physical time domain, in order to solve the discretized time domain sequentially (as required for dynamic stability of the re-formulated model in pseudo-time). The PSA process model was incorporated into a steam-methane reforming process flowsheet, which also featured a highly detailed, spatially distributed model of the reforming furnace. The proposed framework allows the integration of steady-state and dynamic (cylic) models, and the simultaneous optimization of the detailed reactor, PSA process, and flowsheet specifications enables full consideration of process tradeoffs during design optimization.

Chapter 15

Optimal Demand Response Scheduling of an Industrial Air Separation Unit Using Data-Driven Dynamic Models[†]

This chapter presents a methodology for creating low-order, scheduling-relevant models of a multi-product industrial ASU using real-world historical operating data. In the context of demand-response operation, detailed physical models are challenging to use in production scheduling, owing to the need for fast decisions (Chapter 12). To this end, I identify relatively simple dynamic models for economically relevant variables (production rates, power consumption) and the process variables found to limit the dynamic agility of the process and its ability to "respond" to fast-changing electricity price signals. I incorporate the scheduling-relevant models into a DR scheduling problem formulated as a dynamic optimization and present extensive case studies to examine the benefits of DR operation.Although re-scheduling is not considered, the identified low-order models allow the scheduling problem to be solved fast enough for online scheduling [103, 195] and can be easily updated if the controller and/or the plant are modified. The presentation in this chapter follows closely the material published in Tsay et al. (2019) [258].

[†]The contents of this chapter are largely based on the following publication: C. Tsay, A. Kumar, J. Flores-Cerrillo, and M. Baldea. Optimal demand response scheduling of an industrial air separation unit using data-driven dynamic models. Comput. Chem. Eng., 126:22–34, 2019. C.T. is the primary author of the manuscript.

15.1 Background and Motivation

In the field of chemical processing, cryogenic air separation units (ASUs) have been identified as a prime candidate for DR operation, as they consume large amounts of electricity. Air separation units are treated as utility suppliers by their downstream customers (see the discussion of commodity chemicals in Chapter 2), who may place explicit expectations on quality and availability of products (oxygen, nitrogen, and argon), but are otherwise agnostic to time of production. "Load shifting" behavior can be achieved by ramping up production during periods of low electricity prices and storing excess products as cryogenic liquids, which are then vaporized to satisfy gas demands during high-price periods [256]. Traditionally, production scheduling and process control for ASUs (and chemical processes in general) operate as distinct, hierarchical decision-making layers owing to a separation of time scales. However, as described in Chapter 12, this separation is often impossible in the context of DR scheduling, due to the fast-changing nature of electricity markets.

Computing schedules for ASU processes through dynamic optimization with detailed first-principles models is typically computationally intractable in practical amounts of time [65]. To examine the feasibility of DR production-rate modulation in ASUs, Cao et al. [43, 45] presented initial results on dynamic modeling and optimization using a first-principles process model by restricting the time horizon to individual production rate transitions. The same authors subsequently examined alternative approaches for the collection and usage of liquid storage using collocation-based dynamic models [46]. More recently, Schäfer et al. [222] applied a compartmentalization-based model reduction approach for ASUs and demonstrated improved computational performance compared to collocation-based approaches.

In an effort to balance computational complexity with capturing (some of) the relevant process dynamics in production scheduling, many works assume quasi-stationary modes

of operation with additional constraints tailored to reflect the transition capabilities of the plant and control considerations [293, 294, 301]. Zhang et al.[294] extended the mode-based scheduling framework presented by Mitra et al. [170, 169] by introducing surrogate subprocess models that allow for computationally efficient scheduling of continuous process networks. Misra et al. [168] used a state-task network to model production constraints. With a similar motivation, Zhou et al. [301] defined a set of ASU operating modes from historical data, and used convex hulls to reflect feasible operating points and identify the optimal production schedule. Zhao et al. [300] proposed a state-transition network model for scheduling ASUs similar to that of Basán et al. [20], and applied it to the simultaneous scheduling of two multi-product ASUs. Obermeier et al. [186] defined a mode-based scheduling approach to examine relationships between DR scheduling and equipment fatigue.

On the other hand, simplified (e.g., by reducing dimensionality) dynamic models can be employed for computational tractable scheduling as an alternative to assuming quasi-stationary modes. Dias et al. [67] showed that model predictive control (MPC) can be integrated in production scheduling calculations with low computational effort using a nested decision-making structure [306]. A simulation-optimization framework was used, where simple state-space models derived via system identification were employed to predict the open-loop process dynamics. Note that economic MPC can directly incorporate the economic objectives usually considered in scheduling into the control layer [18]; however, optimizing the (large-scale) process model in real-time and considering a time horizon relevant to production scheduling requires some measure of compromise, e.g., a sub-optimal update approach [48].

15.1.1 Description of the Industrial Air Separation Unit Under Consideration

Air separation units (ASUs) separate the components of air predominantly for use in other manufacturing processes. For example, argon is used in welding, oxygen in steel production, and nitrogen in food and metals processing. Purified oxygen also has medical applications. Although several air separation technologies are available, the bulk of high-purity, high-volume industrial gas production is dominated by cryogenic ASUs, which consume immense amounts of energy and primarily rely on electric compressors to handle and compress air feed streams. The industrial cryogenic ASU considered in this work produces five products: liquid nitrogen (LN_2), gaseous nitrogen (GN_2), liquid oxygen (LO_2), gaseous oxygen (GO_2), and liquid argon (LAr). The process operates under model predictive control (MPC) [229], with the multi-variable control system having 12 manipulated variables and 35 controlled variables.

Most works, especially those that explicitly consider process dynamics, typically involve plants that can be appropriately described with first-principles or empirical models of lower dimension. For instance, many previous studies [43, 45, 67, 196, 222] focusing on ASU scheduling involve a simulated plant that produces only nitrogen. Multi-product ASUs have in some instances been considered, but using reduced-order models [46] or assuming quasi-stationary operating points/modes [294, 168]—in essence, another dimensionality-reduction device. One should recognize that such model reduction measures are computationally efficient, and have even allowed extending production scheduling calculations to sites that are served by several multi-product ASUs [300, 301]; however, to my knowledge, there are no previously published results on the explicit incorporation of process dynamics in scheduling calculations for multi-product ASUs.

The flowsheet of the process considered in this work is shown in Figure 15.1. Inlet air

enters the feed compressor (FC), which is driven by a large electric motor. Impurities such as water and hydrocarbons are removed in the prepurifier (PP). The air stream is then cooled in the multistream heat exchanger (MHEX) against warming cryogenic product streams and enters the high-pressure lower column (LC), where it is separated into nearly pure nitrogen "shelf gas" (at the top of the LC) and an oxygen-rich bottoms stream. A portion of the shelf gas provides the reboiler duty to the low-pressure upper column (UC) before being returned to the LC as reflux, while the rest of the shelf gas passes through the MHEX and is combined with the nitrogen product of the UC at the liquefier (LQ). The oxygen-rich bottoms stream of the LC passes through the nitrogen superheater (SH) before being used as a condensing utility for the argon column (ARC) and passed to the UC.

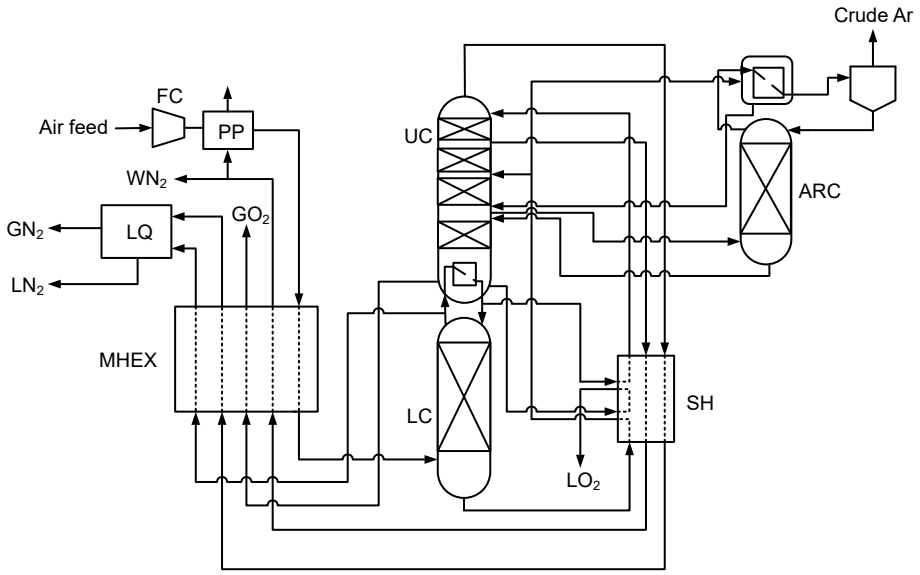

Figure 15.1: Diagram of the industrial air separation unit.

A portion of the combined nitrogen streams is liquefied in the liquefier (LQ), which provides both gaseous and liquid nitrogen products. A "waste" stream of nitrogen is drawn at an intermediate stage close to the top of the UC in order to provide additional refrigeration in the MHEX. The waste nitrogen stream is then vented to the atmosphere. Purified oxygen

product is collected at the bottom of the UC. A portion of the oxygen product is vaporized through the MHEX, while the remainder is passed through the SH. The air fed to the process contains a small amount of argon, which accumulates on an intermediate stage close to the bottom of the UC. At this location, an intermediate vapor stream is drawn and fed to the argon column (ARC). The ARC operates at a very high reflux ratio, and the "crude" argon stream removed at the top of the ARC has a very small flow rate. High purity argon is then obtained in a separate device that is not considered here.

15.1.2 Scheduling Under Dynamic Constraints

Historically, production scheduling calculations relied on the simplifying assumption that the transition of a chemical process from one operating point to another can be defined in terms of a transition period, and that the process predominantly operates at the steady states corresponding to the aforementioned operating points. Under the assumption that the durations of the transitions periods are known, are short compared to period of steady-state operation, and do not change in time, scheduling calculations can effectively ignore the dynamics of the process, and scheduling decisions can be separated from control/operational decisions [229]. This is conceptualized in the hierarchical paradigm shown in Figure 12.1.

For DR scheduling of ASUs, these assumptions may no longer be valid, given that electricity prices change at frequencies similar to the slowest dynamic modes of the process. Figure 15.2 illustrates a hypothetical example, where, in schedule 1, a process violates a dynamic constraint for a particular state variable by making a large step change in operation too quickly. In schedule 2, the schedule accounts for the dynamic agility of the process when making the step change(s). Note the transition times are different in schedules 1 and 2, and accounting for process dynamics may lead to "slower," but dynamically feasible schedules.

The production scheduling problem considering process dynamics and control can

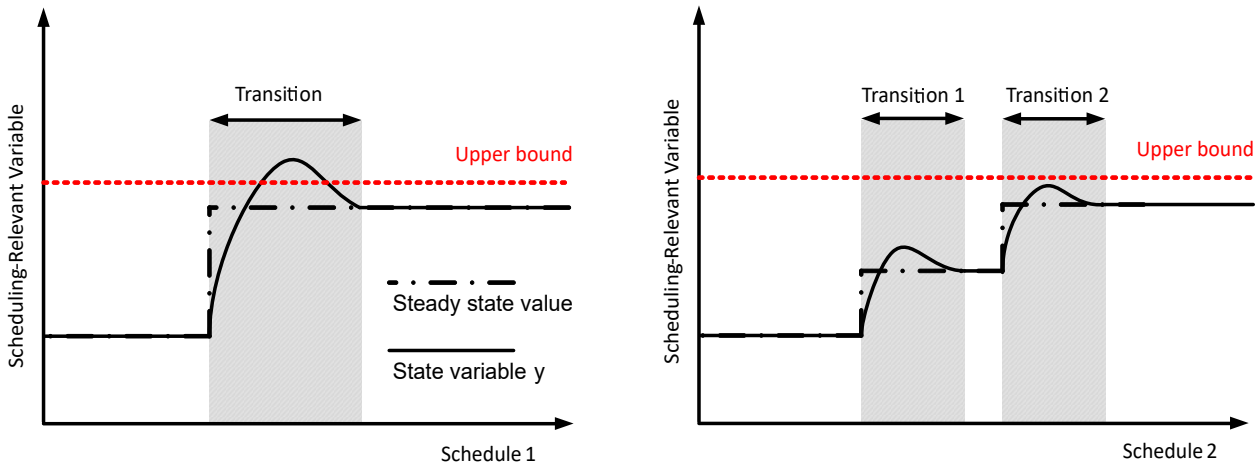

Figure 15.2: Left: Hypothetical process schedule violating a process dynamic constraint during a step change. Right: Hypothetical process schedule with dynamic constraints satisfied.

generally be stated as a dynamic optimization problem, given by (12.6)–(12.11). While formulating the scheduling problem as a dynamic optimization allows for enforcing path constraints, the dynamic optimization is difficult to solve quickly for use in practical situations. Recent works have also highlighted the importance of online re-scheduling [103], further emphasizing the desire for computational expedience. Given the above, the process closed-loop dynamic model, embedded in the dynamic optimization problem as the process model ($\boldsymbol{f}(\cdot)$ and $\boldsymbol{g}(\cdot)$) and the control policy ($\boldsymbol{K}$), should preferably be low-dimensional [16]. The replacement of detailed first-principles models with low-order models, termed "scale-bridging models" or SBMs, for scheduling applications has been the subject of several recent works: the reader is referred to [16, 196] for a detailed discussion. In the following section, I describe the development of SBMs relevant to the industrial problem considered here.

15.1.3 Scheduling-Relevant Scale-Bridging Models

Recent work has suggested system identification as a means to accurately capture *closed-loop* input-output relationships with SBMs, which can then be naturally included in

process scheduling calculations [74, 16]. Pattison et al. [196] demonstrated these ideas on a simulated, single-product ASU, including a closed-loop scheduling, moving-horizon implementation [195]. The work was later extended by Kelley et al. [129], where the authors expedited the computational solution of the same scheduling problem by using a fully-discretized time domain and an exact MILP reformulation. I exploit this data-driven scale-bridging model (SBM) scheduling framework [196] to schedule a large-scale industrial process under MPC using historical operating data that reflect routine operations.

In the course of the time interval spanned by the data, process operators imposed some changes to the operation of the plant, but the dataset lacks any deliberate excitation that could be construed as being part of a system identification experiment. An SBM is an explicit, low-order representation of the closed-loop input-output dynamics of a process, which can be used in scheduling calculations [74]. The SBM scheduling framework is illustrated conceptually in Figure 15.3, along with a comparison with the traditionally separate scheduling and control layers. In the latter, the process operates under closed-loop control, while the scheduling layer provides the targets given to the supervisory controller. Solving a dynamic optimization problem in this paradigm using a dynamic model (which can be low order [67]) requires a closed-loop simulation to compute the process performance for a given schedule. In contrast, the SBM approach [196] directly models the closed-loop process dynamics. The process states at the end of a schedule in either approach could be fed back to the scheduling layer to naturally enable re-scheduling calculations [195].

There are two broad approaches to deriving low-order dynamic models: model reduction and system identification. *Model reduction* refers to the derivation of a low-order model from a more detailed dynamic process model. This can be performed using, e.g., singular perturbation arguments [15] for system models exhibiting multiple time scales. Alternative techniques such as empirical Gramians [107] and proper orthogonal decomposition [284] can

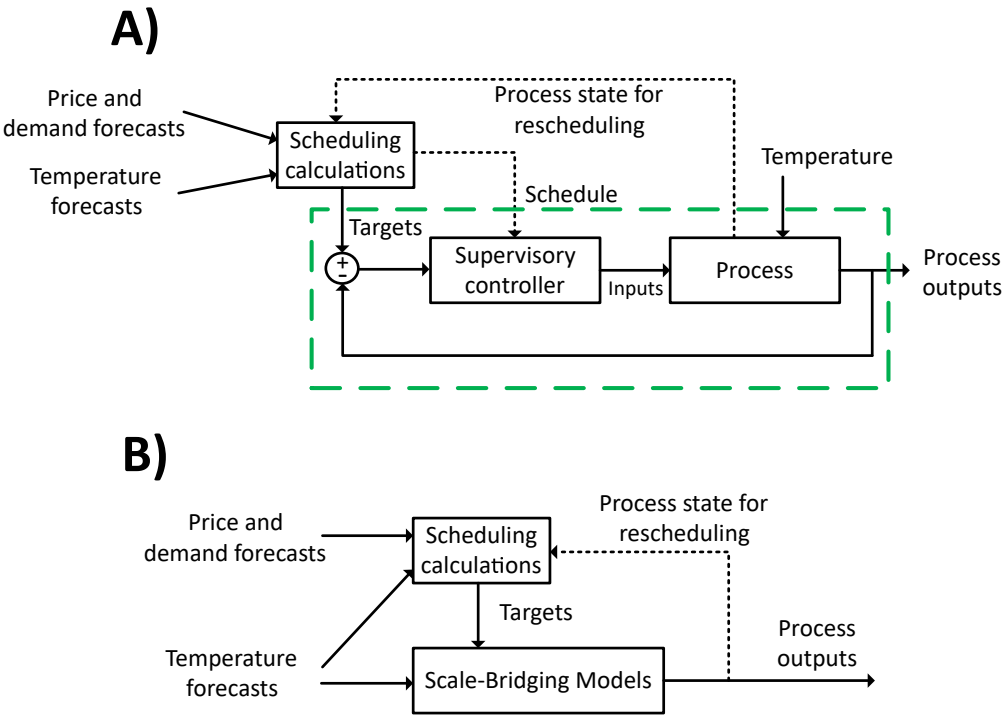

Figure 15.3: A comparison between scheduling using detailed closed-loop process models (A) and scheduling with scale-bridging models (B). The dashed box in (A) depicts the dynamics that are represented by the SBMs.

also be used, although these methods can produce models with states that are not physically meaningful. The second approach, *system identification*, refers to "learning" a low-order process model from operating data; many techniques are available. The interested reader is directed to, e.g., the book by Zhu [302] for an overview of system identification techniques and their applications to process systems. Learning nonlinear process dynamics from data is a highly active area of research [136] and is the approach I take in this chapter.

15.2 Constructing Data-Driven Scale-Bridging Models of the Industrial ASU

15.2.1 System Identification Framework

Chemical processes typically have many sensors that record measurements at frequencies in the order of minutes, generating "big data" sets that can be exploited to understand the process dynamics. Pattison et al. [196] observed that the set of scheduling-relevant variables can be a small subset of the process variables, reducing the dimensionality of the scheduling problem. In particular, they suggested that only i) variables relevant to the scheduling objective function, demand constraints, and inventory constraints and ii) variables near their bounds during historical static or transient operation should be modeled and included in scheduling optimization calculations. The remaining variables are assumed to not be relevant to optimal scheduling calculations, as historical data suggest that they do not hinder process agility. Briefly, the approach for learning SBMs comprises:

(i) Obtaining historical process operating data

(ii) Identifying scheduling-relevant variables to be modeled

(iii) Determining model forms and fitting dynamic models

On the input side of the models, while the process operates under model predictive control (MPC) with six operator set points (targets), I found through trial-and-error that the production rates and other scheduling-relevant state variables could be accurately modeled using only two of the six operator set points, which I will call $[SP1, SP2]$. This finding suggests that the remaining four operator set points either have negligible impact on the variables of interest for scheduling, or that they have historically been set such that they are strongly correlated with the first two set points. On the other hand, I found that ambient

temperature T, which is a measured disturbance variable, has a large effect on the process variables and must be included as an input to the SBMs. Given these findings, I identified multiple-input, single-output (MISO) SBMs. The SBM to predict the behavior of each scheduling-relevant variable was modeled with three inputs, $[T, SP1, SP2]$.

While previous works [67, 196] relied on relatively small *simulated* datasets (whose horizons span a few days to a week) for system identification, in this work I use a large set of *actual* process historical operating data. Since in the present application the data lack the deliberate excitation imposed during system identification experiments, I included 110 days of historical operating data from the industrial process to ensure that sufficiently rich information is available for identifying the desired SBMs. The data were recorded at one-minute intervals by the process historian, during periods of regular, unforced operation. Periods of start-up, shut-down, and process or measurement faults were excluded from this study. These periods were identified easily in the historical data, as the sensors are either off or producing readings that are, e.g., outside the physical bounds for the respective variables. Although shut-downs of the entire ASU could potentially be scheduled to avoid price peaks, they were not considered in this study. However, the shutdown of the *liquefier* was considered, as described later.

The data correspond to 110 days of summer operation, as the summer months tend to have larger fluctuations in electricity prices. The first 88 days (80%) of the data were selected as training data, with the remaining 22 days (20%) used as test data to evaluate the quality of the identified SBMs. While the choice of the functional form of data-driven model can be arbitrary, due to the lack of deliberate system excitation, I chose to identify SBMs in the Hammerstein-Wiener (HW) form. This choice was supported by successful application of HW models to capture the relevant process dynamics in a similar, but smaller-scale application [196]. HW models are structured, requiring fewer parameters to be regressed.

They comprise a linear dynamic component flanked by static, nonlinear input and output transformations. They can be formulated for multiple inputs and a single output (MISO) with the linear dynamic component represented as a state-space model:

$$h_i = \Phi_i(u_i) \tag{15.1}$$

$$\dot{\vec{z}}_i = A\vec{z}_i + Bh_i \tag{15.2}$$

$$y_i = C\vec{z}_i \tag{15.3}$$

$$w = \Psi(\sum y_i) \tag{15.4}$$

where Φ and Ψ are, respectively, the Hammerstein and Wiener blocks corresponding to the static, nonlinear input and output transformations. A, B, and C are the matrices defining the linear state-space dynamical system, u_i denotes the i^{th} model input (for the ASU, $u_i \in [T, SP1, SP2]$), and w is the system output. The linear state-space system is of order $n_{d,i}$, with $\vec{z}_i \in \mathbb{R}^{n_{d,i}}$. The structure of a MISO Hammerstein-Wiener model with three inputs and one output is shown in Figure 15.4.

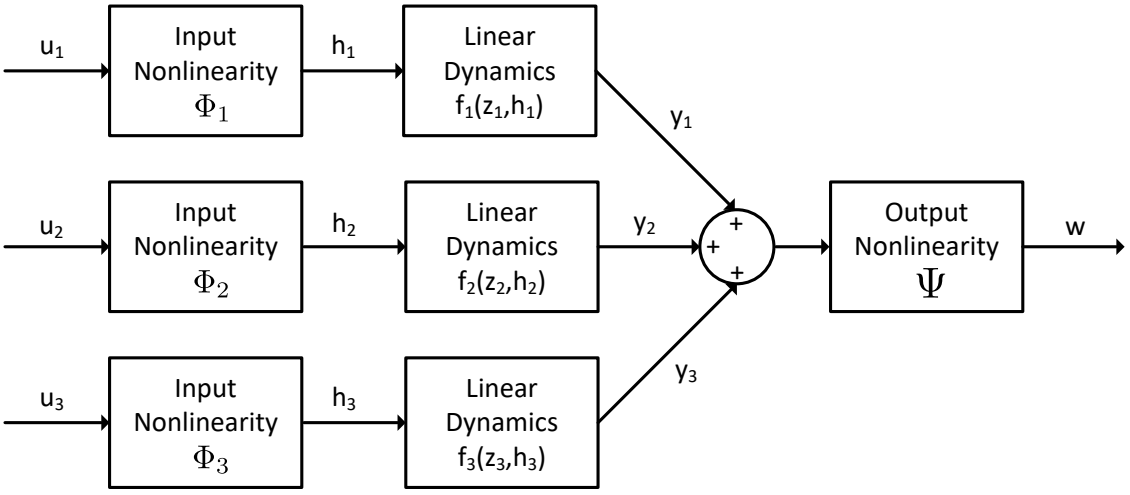

Figure 15.4: MISO Hammerstein-Wiener model with three inputs.

Although the industrial ASU produces five products, the liquid oxygen production rate was relatively constant throughout the operating period represented in the data. As actual industrial data were used in this study, the argon production rate was not disclosed and modeled for confidentiality reasons. The production rates that are therefore considered in DR scheduling calculations (and whose dynamic responses to changes in $[T, SP1, SP2]$ are modeled) are thus liquid nitrogen (LN_2), gaseous nitrogen (GN_2), and gaseous oxygen (GO_2). The MPC implementation includes constraints on several process variables, but I found that only two controlled process variables, denoted here as $CV1$ and $CV2$, closely approach their bounds during operational transitions, indicating they are the primary factors limiting the dynamic agility of the ASU. In the interest of protecting industrial intellectual property, their true nature cannot be disclosed. Together with the power consumption of the plant, the scheduling-relevant outputs of the plant are $w \in [F_{LN2}, F_{GN2}, F_{GO2}, CV1, CV2, W_{total}]$.

15.2.2 System Identification Results

A MISO SBM in Hammerstein-Wiener form (15.1)–(15.4) was identified for each scheduling-relevant variable, and the pertinent details are shown in Table 15.1. The models were obtained using the System Identification Toolbox in MATLAB. Kelley et al. [129] found that the dynamics of an ASU could be accurately represented using Hammerstein-Wiener models with piecewise linear input and output functions, and therefore, input and output nonlinearities were all represented as piecewise linear functions of the form:

$$\Phi_i(u_i) = \frac{pw_{j+1} - pw_j}{bp_{j+1} - bp_j}(u_i - bp_j) + pw_j; \qquad u_i \in [bp_j, bp_{j+1}) \tag{15.5}$$

$$\Psi(\sum y_i) = \frac{pw_{j+1} - pw_j}{bp_{j+1} - bp_j}(\sum y_i - bp_j) + pw_i; \qquad \sum y_i \in [bp_j, bp_{j+1}) \tag{15.6}$$

where $j = 1, ..., n_p + 1$ is the set of piecewise linear segments. The piecewise linear functions are each parameterized by n_p breakpoints that define the connection of adjacent line

Table 15.1: Details of Hammerstein-Wiener scale-bridging models (refer to Figure 15.4 for the model structure).

Output variable	n_p for inputs $[T, SP1, SP2]$	n_d for inputs $[T, SP1, SP2]$	n_p for output	Training NMSE (%)	Test NMSE (%)
F_{LN2}	[2,1,3]	[1,3,2]	5	83.04	74.53
F_{GN2}	[4,3,3]	[3,2,1]	5	76.17	61.52
F_{GO2}	[3,3,2]	[4,3,3]	3	66.39	65.55
W_{total}	[4,3,3]	[3,3,1]	5	82.18	82.67
$CV1$	[2,2,2]	[3,3,3]	4	67.53	61.86
$CV2$	[4,4,2]	[2,3,2]	4	42.22	42.65

segments: bp_i is the value of the breakpoint at segment i, and pw_i is the value of the input function at the breakpoint bp_i. Each piecewise linear transformation has n_p linear segments. To determine the number of piecewise-linear segments n_p for each transformation, the normalized Akaike information criterion (nAIC) was minimized while using a large number of segments for all other piecewise-linear transformations and a high-order linear state-space model. The order of each linear state-space model was similarly determined using the nAIC. Table 15.1 also reports the normalized mean square error (NMSE) for each model on training and test data, defined as:

$$\text{NMSE} = 1 - \frac{||x_{ref} - x||}{||x_{ref} - \text{mean}(x_{ref})||} \tag{15.7}$$

The SBM predictions for all six variables listed in Table 15.1 for a week within the training dataset are plotted in Figure 15.5. The low NMSE values are largely due to high-frequency dynamics in the historical data: the MPC operates (and the data are recorded) on a 1-minute interval, while the operator set points are changed much less frequently. The model predictions are close to the time-averaged behavior of the process variables in most cases, and the accuracy of model predictions could most likely be improved in future studies by collecting process (or simulated) data with higher excitation levels or conducting system identification experiments. Additional data may also enable improving model accuracy by

affording the selection of a different model representation/structure. HW models are inherently restricted to a linear representation of the dynamics; although this feature does not pose limitations for this study, it may become limiting in cases where the dominant time constant of the process changes significantly over time or as a result of changes in operating regime. Note that all output and input variables listed or plotted in this article have been scaled and filtered to preserve the confidential nature of industrial data.

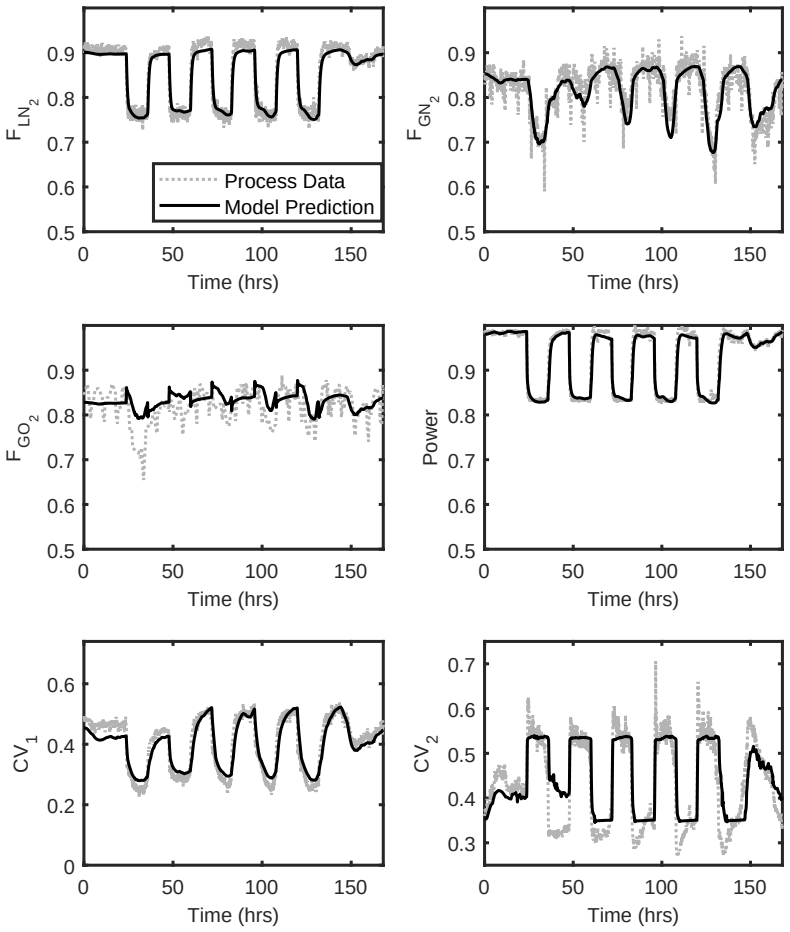

Figure 15.5: Scale-bridging model predictions for a week of test data.

15.3 Optimal Demand Response Operation of the ASU

The continuous-time ASU demand-response scheduling problem was formulated as a dynamic optimization, using the identified SBMs to predict the transient behavior of relevant process variables. The objective is to minimize the cost of electricity purchased from the grid over a four-day horizon by manipulating the trajectories of the two operator set points. Constraints are added such that product demands are met and that the scheduling-relevant CVs remain within their prescribed bounds. The disturbance variable, ambient temperature, is also accounted for. For this initial study, I assume that, (i) production constraints are known and can be enforced on total production, (ii) the storage capacity is sufficient to accommodate products to cover the demand for the four-day period considered, and (iii) perfect forecasts of ambient temperature and electricity price are available for the entire scheduling horizon. While these assumptions may not hold true in practical application, the approach provides valuable insight into the maximum benefit that can be derived from DR scheduling of industrial ASUs. The DR scheduling optimization problem can be written as:

$$\min_{SP1(t),SP2(t)} \quad \int_{t=0}^{t=t_f} \mathcal{P}(t) W_{total}(t) dt \tag{15.8}$$

$$s.t. \quad \dot{\mathbf{x}}^{HW} = \boldsymbol{f}^{HW}(SP1, SP2, T) \tag{15.9}$$

$$CV1^L \leq CV1 \leq CV1^U \tag{15.10}$$

$$CV2^L \leq CV2 \leq CV2^U \tag{15.11}$$

$$\int_{t=0}^{t=t_f} F_{LN2}(t) dt \geq D_{LN2} \tag{15.12}$$

$$\int_{t=0}^{t=t_f} F_{GN2}(t) dt \geq D_{GN2} \tag{15.13}$$

$$\int_{t=0}^{t=t_f} F_{GO2}(t) dt \geq D_{GO2} \tag{15.14}$$

where $\mathcal{P}(t)$ is the time-varying price of electricity, $\boldsymbol{f}^{HW}$ denote the identified Hammerstein-Wiener models, and D_i is the (known) demand for product i over the four-day window.

(15.10) and (15.11) are path constraints on the scheduling-relevant CVs, and (15.12)–(15.14) are production constraints on the ASU products.

A four-day period from the test data, where operator set point values were constant in time, was identified as a base case. Extended periods of operation with constant operator set points in the historical data occurred only with the plant running at maximum capacity (maximum here refers to the highest value present in the available dataset). In order to simulate over-production without extrapolating, the same four-day period was simulated for operation at 95% (in terms of power consumption) of the plant capacity. Historical ambient temperature profiles were obtained from the process historian, and historical electricity prices supplied by the regional ISO (independent system operator) were used. I note that, although the operator set points are held constant in the base case, the process states are still affected by fluctuations of the disturbance variable (temperature) in time. The real-time and day-ahead prices over the selected four-day horizon are shown in Figure 15.6. The real-time electricity prices are subject to larger variations, with a sharp peak of over 500 $/MWh in the second day, but also some periods of negative electricity price. In contrast, the day-ahead electricity prices follow a "smoother" periodic pattern.

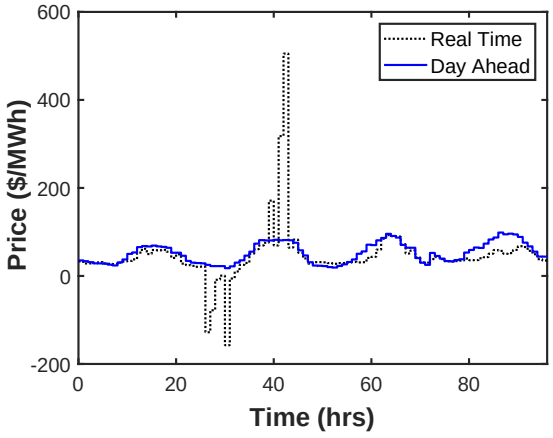

Figure 15.6: Electricity prices over the selected four-day horizon.

The models were implemented in gPROMS [204], and the optimal scheduling calculations were performed therein using a sequential dynamic optimization solver. An alternative approach could involve full discretization of the time domain and solution of the resulting large-scale MILP [129]. The schedules reported herein represent the local optima found using the base case as the initial guess. The calculations were performed on a 64-bit Windows system with Intel Core i7-8700 CPU at 3.20 GHz and 16GB RAM. Although only the offline scheduling problem was considered in this work, the optimization computations required about 10 minutes each, making the method amenable to online scheduling and re-scheduling calculations. The reader is referred to the papers by Pattison et al. [195, 196] for details on the formulation of the optimal scheduling problem under dynamic constraints represented as HW models, and to the work of Vassiliadis et al. [268, 269] for an in-depth discussion on sequential dynamic optimization.

15.3.1 Results for the Case of Real-Time Electricity Pricing

The optimization problem in (15.8)–(15.14) was solved using the prices $\mathcal{P}(t)$ given by the historical real-time electricity price shown in Figure 15.6. As their name implies, these prices are set in real time and, as a consequence, their values for a 96-hour future time horizon would not be available in practice at the time when scheduling calculations are performed. Rather, predictions would need to be used to generate the optimal schedule. The bounds defined in the MPC system for $CV1$ and $CV2$ were used as their respective upper and lower bounds in (15.10)–(15.11). I will refer to this problem as P1. The same problem was solved using a 10% backoff constraint for both $CV1$ and $CV2$ to generate a more conservative schedule, noting that the SBMs representing process dynamics may not be completely accurate (Table 15.1). Such "backoffs" for active constraints have been implemented for SBM-based scheduling [67, 196], and are in effect implemented in many

practical situations to avoid infeasible operation in the presence of disturbances and/or model inaccuracy (see the works by Narraway and Perkins [178] and Aske et al. [7] for further information). I will refer to the second problem, which includes the aforementioned backoff constraints, as P2. For each variable, the 10% backoff was computed by adding/subtracting 10% of the scaled variable range to/from its lower/upper bound.

Optimization of the operator set points was treated using a control vector parameterization approach [268, 269], using two-hour piece-wise constant profiles. As expected, the optimal schedules found for both P1 and P2 involve over-producing (and increasing power consumption with respect to the base case) when electricity prices are low. Conversely, power consumption is decreased when electricity prices are high. The power demand profiles for P1 and P2 are shown in Figure 15.7. Although the identified SBMs are (piecewise) continuous, a visual comparison of the results against historical data suggests that the sharp drop-off in power consumption between hours 40-45 in both computed schedules corresponds to a liquefier shutdown, where power consumption is drastically decreased at the cost of producing no LN2 ($P_{LN2} \approx 0$).

The LN2 production profiles for the computed optimal schedules are shown in Figure 15.8. In the optimal solution for both P1 and P2, only the LN2 production constraint (15.12) is exactly met, as the other two products are over-produced throughout the four-day window to allow for excess LN2 production. LN2 production is decreased to practically 0 during the liquefier shutdown episodes, as can be easily seen in Figure 15.8. I again note that the variables are scaled and that a value of $F_{LN2} = 0$ actually corresponds to a "negative" net LN2 production rate, i.e., LN2 is transferred from storage to the UC (Figure 15.1) when the liquefier is turned off. The profiles of $CV1$ at the optimal points are also shown in Figure 15.8; $CV2$ did not reach its bounds in the optimal schedules, although the bound is reached in some optimization iterations. It can be seen that using the tighter backoff

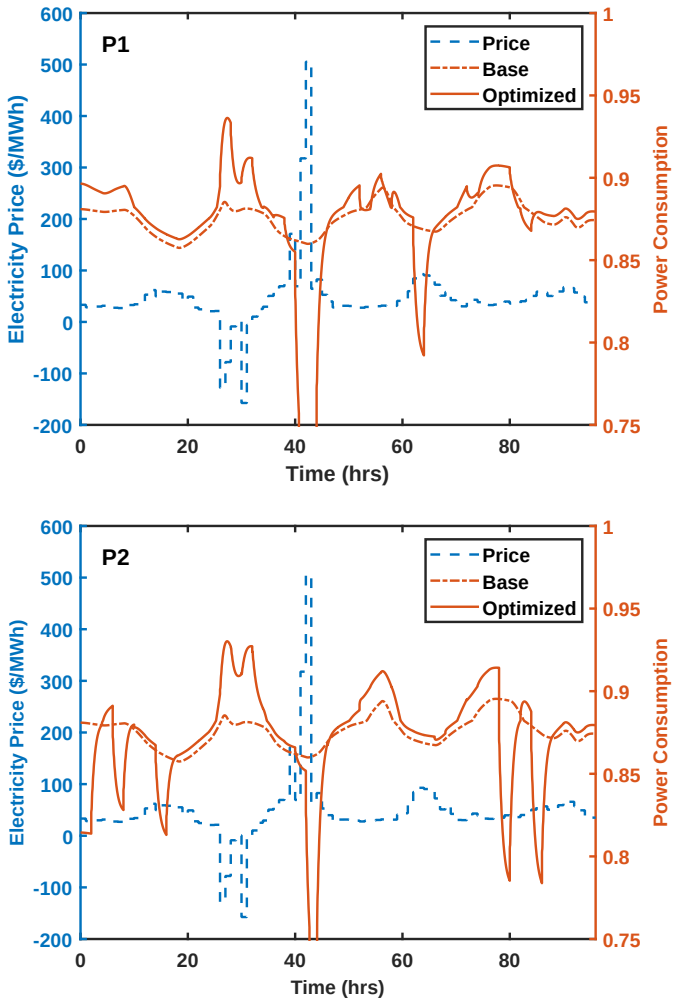

Figure 15.7: Power consumption predicted by SBMs for the optimal schedules in the real-time market using the MPC bounds (P1, top) and backoff bounds (P2, bottom).

constraint results in $CV1$ meeting its lower bound at multiple points in the solution of P2. The backoff constraint limits liquefier shutdowns compared to the case with no backoff, and thus the economic savings are decreased. This demonstrates that the SBM-based scheduling approach is capable of identifying schedules that are dynamically feasible, as given by the process and variable bounds.

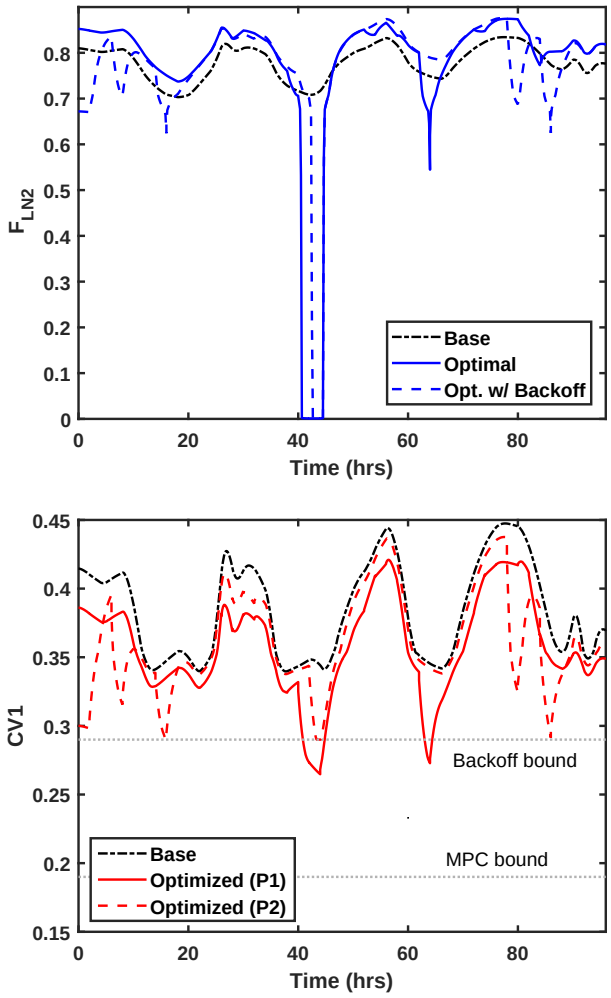

Figure 15.8: Profiles of F_{LN2} and $CV1$ predicted by SBMs for the computed optimal schedules in the real-time market.

I define C^0_{RT} as the incurred cost for the four-day operation period in the base case given the real-time electricity prices. The optimal schedule found by solving P1 results in a 13.1% savings compared to C^0_{RT}, while the schedule found by solving P2 results in a 6.8% savings. While the savings decrease considerably when the dynamic agility of the process is used conservatively (constrained with the 10% backoff), both savings values represent a significant economic advantage in the commoditized industrial gas sector. These large

savings are due to perfect predictions (including of price spikes and negative prices), perfect temperature forecasts, and no storage constraint(s). If large spikes in electricity price are predicted correctly, the ASU can shut down its liquefier and sell LN2 over-produced at other times to exploit its ability to store product (effectively storing energy in the form of molecules). Although no storage constraints were modeled, the required storage tank capacity was estimated by examining the maximum over/under-production quantity of LN2. Using this estimate, the solution for P1 requires a cryogenic liquid storage capacity of 1.92 times the hourly average production, while the solution for P2 requires a tank capacity of 1.68 times hourly average production. Industrial ASUs typically have ample storage capacity available, and I do not expect storage of cryogenic liquids to become a limitation in practical implementations of this DR strategy.

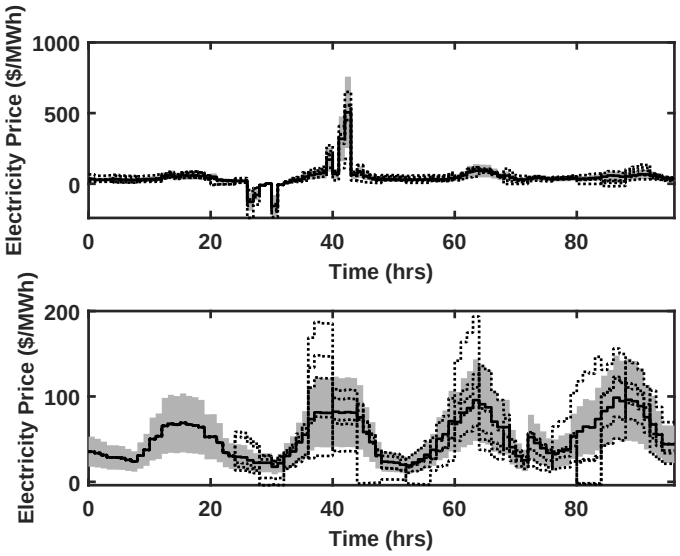

Figure 15.9: Real-time (top) and day-ahead (bottom) electricity prices under uncertainty. The shaded area denotes one standard deviation for $\epsilon_i \sim \mathrm{N}(0, 50\%)$. Five Monte-Carlo (dotted) samples are shown.

A sensitivity analysis can be conducted to explore the impact of imperfect price forecasts. Denoting the electricity price prediction for hour i as $\hat{\mathcal{P}}_j, \forall\ j = 1, ..., 96$, I define

the relative error of the electricity price forecast as:

$$\epsilon_j = \frac{\hat{\mathcal{P}}_j - \mathcal{P}_i}{\mathcal{P}_j} \tag{15.15}$$

where $\mathcal{P}_j$ is the true electricity price in hour j. For the purpose of this study, I assume that ϵ_j are independent and identically distributed, although, e.g., Gaussian process models [139] could give smoother electricity-price prediction profiles by accounting for hour-to-hour error correlations. Figure 15.9 shows the electricity prices for the four-day horizon with $\epsilon_j \sim \mathrm{N}(0, 50\%)$, where $\mathrm{N}(\mu, \sigma)$ denotes the normal distribution with mean μ and standard deviation σ. This relative error representation for real-time pricing is optimistic, as the occurrence and timing of the peak(s) and period(s) of negative price are still predicted properly (in terms of expected value).

To explore the effect of price uncertainty on schedule optimality, Monte Carlo simulations were performed with $\epsilon_j \sim \mathrm{N}(0, \sigma)$. Each Monte Carlo simulation was run with 5000 samples, and uncertainty was coarse-grained into 4-hour blocks to reduce the dimensionality of the uncertainty space ($\epsilon_i = \frac{\hat{\mathcal{P}}_j - \mathcal{P}_j}{\mathcal{P}_j}, \{i = 1, ..., 24; j = 4i - 3, ..., 4i\}$). The first five samples for $\sigma = 50\%$ are shown in Figure 15.9. Generating a Monte Carlo sample involves in fact two simulations: one with the base case set points (which creates the basis for comparison) and one with the optimal schedule. The output datum from each sample simulation is the ratio of the cost incurred by implementing the optimal schedule, denoted as $C_{RT,m}; m = 1, ..., 5000$, to the cost associated with executing the base case schedule, denoted as $C^0_{RT,m}; m = 1, ..., 5000$.

The sampled values of $C_{RT,m}$ and $C^0_{RT,m}$ are approximately normally distributed, as the uncertain variable (electricity price) affects the variable of interest (plant operating cost) linearly. The stochastic relative cost of the optimal schedule, $\frac{C_{RT,m}}{C^0_{RT,m}}$ is thus the ratio of two (approximately) normal distributions. The closed-form expression of the ratio distribution

is complicated [113], and I approximate it using a skewed Gaussian distribution to calculate relevant statistics. The approximation is accurate for low values of σ, but the quality of the approximation deteriorates noticeably for $\sigma \geq 80\%$. Histograms of the relative costs found in the Monte Carlo simulations for P1 and P2 using $\sigma = \{10\%, 30\%, 50\%\}$ are shown in Figure 15.10 (note the different abscissa scalings). There is a non-negligible probability that the relative cost exceeds unity when uncertainty is considered for electricity prices, suggesting that demand-response operation may not always be economically favorable when the accuracy of price forecasts is low. The probabilities of relative cost exceeding 1 are plotted in Figure 15.14. Even with the optimistic assumption that the expected value of electricity price is the predicted value (i.e., times of electricity price peaks and negative electricity prices are properly identified), the optimal schedule(s) may be economically worse than constant set point operation. The distributions shown in Figure 15.10 are skewed right, indicating that although the chances of exceeding constant-operation costs may be low, the probabilities of encountering extreme values are greater on the right-hand side of the distribution for the real-time electricity market.

15.3.2 Results for the Case of Day-Ahead Electricity Pricing

The same optimization problem given in (15.8)–(15.14) was solved using the prices $\mathcal{P}(t)$ given by the historical day-ahead electricity price at the same location (as shown in Figure 15.6). These prices have a smaller range of fluctuation and are relatively more predictable, but are still subject to some uncertainty as they are, in fact, only known a single day (rather than 96 hours) in advance. The scheduling problem was again solved with the bounds specified in the process MPC as the control variable bounds (P3) and with a 10% backoff constraint (P4). The optimal schedules for the ASU involve shutting down the liquefier once in each case, corresponding to times of high electricity prices. The power consumption

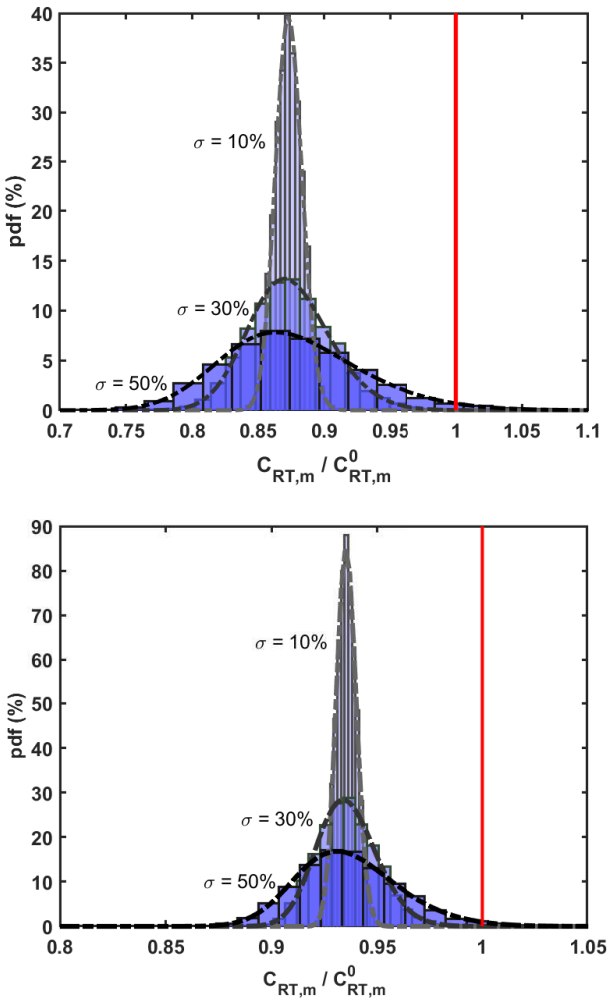

Figure 15.10: Relative costs found in Monte Carlo simulations for P1 (top) and P2 (bottom) using $\sigma = \{10\%, 30\%, 50\%\}$. Skewed Gaussian pdfs are shown as dashed lines.

profiles for both optimal schedules are shown in Figure 15.11. The liquefier shutdown in the solution to P3 occurs during the third electricity price peak, even though the electricity price is marginally higher in the fourth peak, likely reflecting a locally optimal solution.

Due to the scheduled liquefier shutdown periods, LN2 is again the limiting product in the optimal schedules, and the LN2 production profiles are shown in Figure 15.12. The cost savings are primarily due to shutting down the liquefier during times of peak electricity price

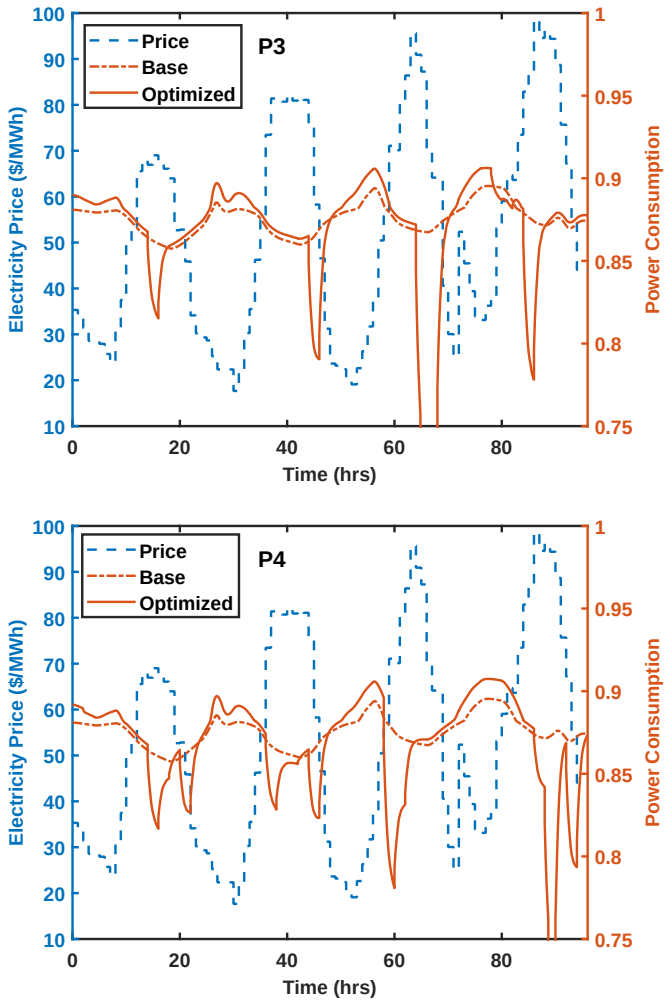

Figure 15.11: Power consumption predicted by SBMs for the optimal schedules in the day-ahead market using the MPC bounds (P3, top) and backoff bounds (P4, bottom).

and over-producing liquid nitrogen at times of lower electricity price. This is likely strongly related to the large number of liquefier shutdown events recorded in the training data set. While this inherently binary decision is handled using the HW model, a discrete variable could also model the on/off state of the liquefier. The profiles of $CV1$ at the optimal points are also shown in Figure 15.12; $CV2$ is again not found to reach its bounds in the optimal schedules. The backoff constraint is met multiple times in the solution to P4.

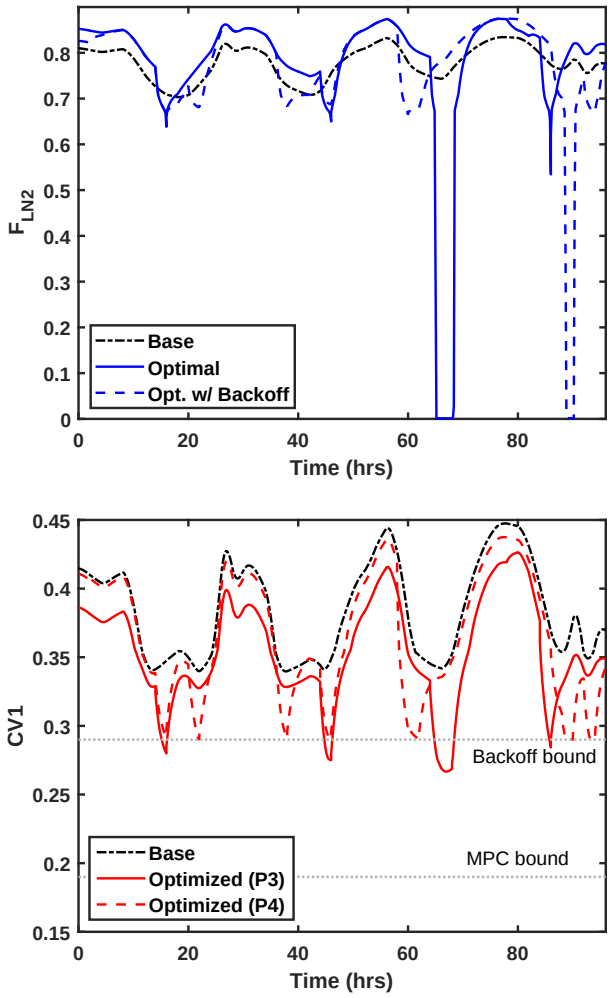

Figure 15.12: Profiles of P_{LN2} and $CV1$ predicted by SBMs for the computed optimal schedules in the day-ahead market.

I again define C^0_{DA} as the incurred cost of four-day operation in the base case given day-ahead electricity prices. The optimal schedule found by solving P3 results in a 3.6% savings compared to C^0_{DA}, while the schedule given by the solution of P4 results in a 2.4% savings. Although lower than in the case of the real-time market, these savings still constitute a significant economic advantage in the industrial gas sector. The economic benefits are lower when the backoff constraint is used, as the tighter bound for $CV1$ gives a more conservative

measure to the degree to which the ASU power consumption can be modulated (Figure 15.12). Using the same storage capacity estimation method as above, the solution for P3 requires a liquid storage capacity of 2.62 times the hourly average production, while the solution for P4 requires a capacity of 1.95 times the average hourly production. These storage capacities are slightly larger than the corresponding values found for the real-time electricity price cases (P1 and P2).

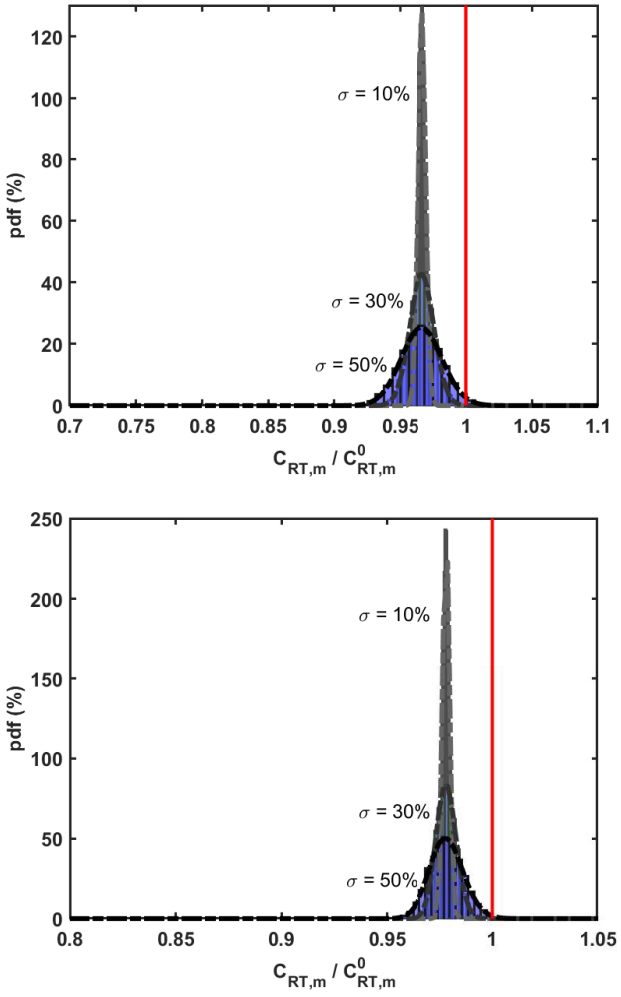

Figure 15.13: Relative costs found in Monte Carlo simulations for P3 (top) and P4 (bottom) using $\sigma = \{10\%, 30\%, 50\%\}$. Skewed Gaussian pdfs are shown as dashed lines.

The effect of price uncertainty on the schedules is again explored using the same Monte Carlo simulation with $\epsilon_i \sim \mathrm{N}(0, \sigma)$, but with uncertainty removed in the first 24 hours (where day-ahead prices would be exactly known). The first five samples for $\sigma = 50\%$ are shown in Figure 15.9. Histograms of the relative costs found in the Monte Carlo simulations for P3 and P4 using $\sigma = \{10\%, 30\%, 50\%\}$ are shown in Figure 15.13. The same abscissae as in Figure 15.10 were used in Figure 15.13 to facilitate comparison between results for the real-time and day-ahead markets. A comparison between Figures 15.10 and 15.13 reveals that, although the expected relative cost of DR operation in the day-ahead market is higher (demand response in the real-time market yields larger savings in the case of no error in the expected values of electricity prices), the schedules based on day-ahead data are less affected by price uncertainty, and the distributions of relative costs found for the day-ahead market have significantly less variance.The probabilities of relative costs exceeding unity, estimated from Monte Carlo frequencies for all four schedules, are shown in Figure 15.14.

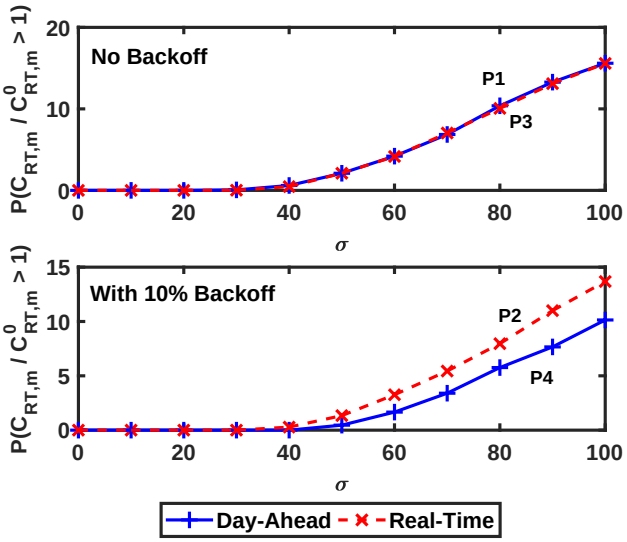

Figure 15.14: Effect of increasing price uncertainty on the probability of exceeding constant-operation cost. Probabilities are estimated by frequency in 5000 Monte Carlo samples.

The mean relative costs and standard deviations for all four schedules are shown in Figure 15.15. Although the mean values of relative costs increase with electricity price uncertainty for DR operation in the real-time market (P1 and P2), they are relatively flat for the schedules created with day-ahead prices (P3 and P4).

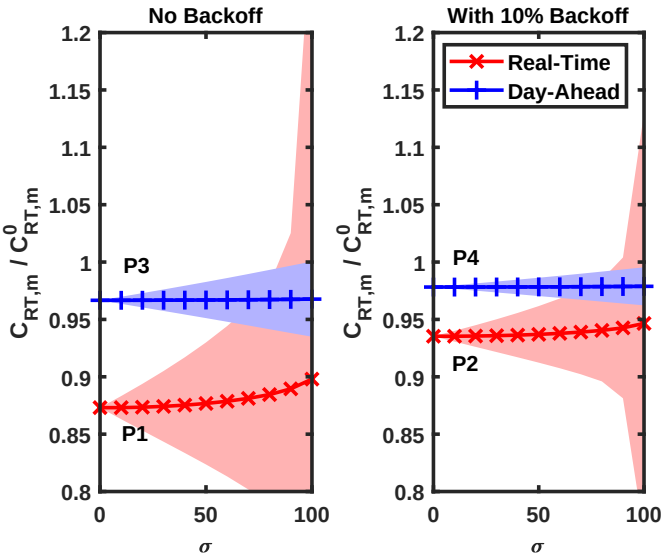

Figure 15.15: Effect of increasing price uncertainty on relative costs of production determined using 5000 Monte Carlo samples. The marked lines depict the mean relative costs, while the shaded regions depict one standard of deviation. Note that the distributions are in fact skewed.

In the cases with no backoff constraint (P1 and P3), the probabilities of relative cost exceeding one are almost identical between day-ahead and real-time markets for all values of uncertainty. However, Figure 15.15 shows that DR scheduling in the real-time market increases the chances of extreme values. For the more conservative schedules generated with a 10% backoff constraint (P2 and P4), the probabilities of relative cost exceeding one are greater in the real-time market, and extreme values are again more likely in the real-time market. These observations confirm the intuition that scheduling in the real-time market is riskier for the same relative uncertainty in electricity prices, although the expected values of

savings are larger. With the given assumption that $E[\hat{\mathcal{P}}_j] = E[\mathcal{P}_j]; \forall\, j = 1, ..., 96$, scheduling in the real-time market produces more savings. Again note that this assumption corresponds to knowing the exact timing of extreme electricity prices (in terms of expected value), which may not be valid in the case of real-time electricity markets. Using the backoff constraint (P2 and P4) results in more conservative schedules, and consequently the standard deviations in relative cost (and probabilities of relative costs exceeding one) are decreased.

15.4 Summary

Cryogenic air separation units (ASUs) are a prime candidate for DR operation, as they consume large amounts of electricity, but their products are agnostic to time of production. However, optimal scheduling calculations for industrial ASUs must account for process dynamics in order to ensure that product quality and process safety constraints are met at all times during the execution of the schedule. This chapter described a data-driven approach to learn the scheduling-relevant dynamics of an industrial process from its historical operations without any deliberate excitation. With these models, the scheduling problem is formulated as a dynamic optimization problem that is easily tractable from a computational point of view. I demonstrated the value of the proposed scheduling framework by comparing the demand response operation of the industrial ASU against a constant operation base case, finding that over 13% cost savings over a four-day horizon can be achieved in real-time electricity markets. In the more conservative day-ahead electricity market, I still found that over 3% cost savings are possible over the four-day horizon.

Chapter 16

Integrating Production Scheduling and Process Control Using Latent Variable Dynamic Models[†]

In Chapter 15, I derived approximations of process dynamics from recorded historical operating data, but selecting the model structure and scope (e.g., the scheduling-relevant variables) required significant engineering expertise. This chapter develops a systematic framework for performing scheduling-oriented modeling and/or model reduction of the dynamics of chemical processes without such expertise. Specifically, I propose deriving low-dimensional, scheduling-relevant dynamic models using a latent-variable representation. I begin by analyzing the intrinsic (approximate) low-dimensionality of closed-loop process dynamics relevant to scheduling. This observation motivates learning the underlying *latent manifold* that describes the process behavior in its intrinsic dimension, and using the transformation to improve the computational performance of scheduling calculations. Using an air separation case study, I show that the approach compares favorably against previous works, and I investigate tuning the dimensionality of the reduced-order representation (i.e., selecting the number of latent variables) to manage the tradeoff between optimization problem size and model accuracy. The presentation in this chapter closely follows the material published in Tsay and Baldea (2020) [255].

[†]The contents of this chapter are largely based on the following publication: C. Tsay and M. Baldea. Integrating production scheduling and process control using latent variable dynamic models. Control Eng. Pract., 94:104201, 2020. C.T. is the primary author of the manuscript.

16.1 Low Dimensionality of Scheduling-Relevant Dynamics

Theoretical developments [15, 16] have shown that the closed-loop, input-output dynamic behavior of process systems (i.e., the response of the process in (12.6)–(12.11) to changes in $\mathbf{y}^{sp}$) may be quite slow in comparison to the evolution of states of the individual process units. The former, input-output behavior often evolves over time scales relevant to process scheduling calculations, particularly in the context of the fast-changing market conditions. Moreover, the aforementioned results suggest that the input-output dynamic behavior can be described, or at least usefully approximated, using a low-order model of dimension much smaller than that of the state variable vector $\mathbf{x}$. On this basis, Chapter 15 represented the dynamics of an ASU in scheduling calculations using time scale-bridging models (SBMs), which are a low-order approximation of the closed-loop dynamics of the process.

In effect, Chapter 15 reduced the dimensionality–and, consequently, the computational complexity—of the scheduling problem (12.6)–(12.11) by restricting modeling efforts to "scheduling-relevant variables." In particular, a subset $\mathbf{w} \subseteq [\mathbf{x}, \mathbf{y}, \mathbf{u}]$ is defined, that includes the input and output variables ($\mathbf{u}$ and $\mathbf{y}$) that affect $J(\cdot)$ and $l(\cdot)$. Process state and output variables ($\mathbf{x}$ and $\mathbf{y}$) whose constraints are active during steady-state operation or during process transitions are intuitively assumed to limit the process dynamic agility and are also included in $\mathbf{w}$ to ensure that the resulting schedules do not violate any constraints (i.e., guarantee the dynamic feasibility of a schedule). In essence, the dimensionality reduction occurs through a heuristic and expertise-intensive selection of the scheduling-relevant variables $\hat{\mathbf{x}} \subseteq \mathbf{x}$, $\hat{\mathbf{y}} \subseteq \mathbf{y}$, and $\hat{\mathbf{u}} \subseteq \mathbf{u}$. The resulting scheduling optimization problem is similar

to (12.6)–(12.11), but can contain significantly fewer variables and constraints:

$$\max_{\mathbf{y^{sp}}} \quad \int_{t=0}^{t=t_f} J(\mathbf{w},t)dt \tag{16.1}$$

$$s.t. \quad \dot{\mathbf{w}} = \begin{bmatrix} \dot{\hat{\mathbf{y}}} \\ \dot{\hat{\mathbf{x}}} \\ \dot{\hat{\mathbf{u}}} \end{bmatrix} = \boldsymbol{f}^{SBM}(\mathbf{w}, \mathbf{y}^{sp}) \tag{16.2}$$

$$l(\mathbf{w},t) = 0 \tag{16.3}$$

$$\mathbf{w}^L \leq \mathbf{w} \leq \mathbf{w}^U \tag{16.4}$$

where $l(\mathbf{w},t)$ may include both path and endpoint storage/demand constraints. The reduced-space dynamic model $\boldsymbol{f}^{SBM}(\mathbf{w},\mathbf{y}^{sp})$ represents the scale-bridging model (SBM), which approximates the closed-loop, input-output relationships between the process setpoints and the scheduling-relevant variables, e.g., w in (15.1)–(15.4). The dynamic model $\boldsymbol{f}^{SBM}$ replaces the process model present in (12.6)–(12.11).

Dimensionality Reduction via Manifold Learning. While limiting the scheduling problem to only consider the dynamics of *scheduling-relevant* variables ($\mathbf{w}$) can be an effective form of dimensionality reduction, the selection of these variables relies on human input, insight, and expertise. On the other hand, chemical processes generate "big data" sets (Chapter 15) that can be exploited to understand the underlying system behavior. There exist many approaches for learning low-dimensional representations of a dynamical system from recorded data. In this context, *manifold learning* refers to identifying a low-dimensional manifold on which higher-dimensional data points intrinsically lie. The learned manifold represents a subspace of the full-dimensional variable space that explains (most of) the variation observed in the data set. Observations of the original system can be transformed to (projected on) a smaller set of *latent variables* that parameterize the manifold.

A broad class of unsupervised machine learning algorithms can be applied to the task of manifold learning. Pearson [199] introduced principal component analysis (PCA) in 1901,

and the technique is now a widely accepted dimensionality-reduction technique, especially for data related to linear systems. PCA consists of finding a linear coordinate transformation whereby the data are projected on a new set of latent variables such that the amount of variance captured by each successive latent variable, or "principal component," is maximized. Latent variables based on linear combinations of the original variables, e.g., from PCA, are commonly used in the process industries for process monitoring and troubleshooting [161]. They have also found applications in process control, where they can be employed to reduce the dimension of the controlled variable space. For example, latent variables can replace the original process controlled variables to simplify controller calculations [89, 148].

While PCA is limited to finding linear mappings, a number of nonlinear manifold learning algorithms have been presented. A simple nonlinear extension of PCA is kernel PCA, where a nonlinear kernel is first applied, and PCA is performed in the processed feature space [226]. Several researchers have studied the relationships between PCA and a particular class of artificial neural network known as autoencoders. Of particular note, Sanger [218] showed that linear autoencoders correspond exactly to PCA, while Kramer [141] proposed nonlinear autoencoders as a form of generic nonlinear PCA. Many other nonlinear manifold learning techniques have been since proposed; the reader is referred to the book by Lee and Verleysen [150] and the review by Van Der Maaten et al. [266] for further details.

16.2 Scheduling with Learned Latent Variables

I observe now that the *intrinsic dimensionality* (the number of independent variables underlying the significant nonrandom variations in the observations) of the closed-loop behavior of a chemical process can be much lower than the apparent *extrinsic dimensionality* $(n+m+u)$. In particular, the dimensionality of the process input $\mathbf{u} \in \mathbb{R}^u$, output $\mathbf{y} \in \mathbb{R}^m$,

and state $\mathbf{x} \in \mathbb{R}^n$ variables is linked to the process state variables $\mathbf{x}$:

$$\mathbf{x}^* := \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \\ \mathbf{u} \end{bmatrix} = \begin{bmatrix} \mathbf{x} \\ \boldsymbol{h}(\mathbf{x}) \\ \boldsymbol{K}(\mathbf{y}^{sp} - \boldsymbol{h}(\mathbf{x})) \end{bmatrix} \tag{16.5}$$

where I define the augmented process state variable vector $\mathbf{x}^*$ as $[\mathbf{x}, \mathbf{y}, \mathbf{u}]$. Therefore, the mapping relating the setpoints/targets set by the scheduling layer, $\mathbf{y}^{sp}$, to $\mathbf{x}^*$ has an intrinsic dimensionality equal to $\dim_i(\mathbf{x})$. I use $\dim(\cdot)$ to denote extrinsic dimensionality and $\dim_i(\cdot)$ to denote the intrinsic dimensionality, as defined above. Note that this assumes that an explicit function $\mathbf{y} = \boldsymbol{h}(\mathbf{x})$ can be derived from $\boldsymbol{g}(\mathbf{x}, \mathbf{y}, \mathbf{u}, \mathbf{d}, \boldsymbol{\theta}, t) = 0$ in (12.6)–(12.11). Moreover, this analysis assumes that $\mathbf{u} = \boldsymbol{K}(\mathbf{y}^{sp} - \boldsymbol{h}(\mathbf{x}))$ can be evaluated directly, i.e., an explicit control law exists. In the case of an implicit/optimization-based controller, an explicit relationship may still exist, or an approximation may be possible [202]. I will examine the case of dimensionality reduction for a process operating under an optimization-based controller in the case study presented later in this chapter.

Remark 16.1. For the particular case of model predictive control (MPC), Lovelett et al. [157] observed that *intrinsic* dimensionality of the optimal control policy $\mathbf{u}(t)$ may be significantly lower than its *extrinsic* dimensionality—an observation aligned with the aforementioned findings concerning the process dynamics. This may occur in several circumstances, such as when the dynamics of the system itself lie on a low-dimensional, slow manifold (as mentioned above) [15], or where the state-space realization is non-minimal order (i.e., it contains redundant information [157]).

In this work, I propose a new learning-based approach for low-order SBM generation, involving a latent manifold mapping of the augmented process state variable vector $\mathbf{x}^*$. I seek an invertible mapping $\mathbf{x}^* \leftrightarrow \boldsymbol{\phi}$, with $\boldsymbol{\phi} \in \mathbb{R}^p$, denoted as $\boldsymbol{\phi} = \boldsymbol{c}(\mathbf{x}^*)$ and $\mathbf{x}^* = \boldsymbol{c}^{inv}(\boldsymbol{\phi})$. Furthermore I desire to identify the mapping $\boldsymbol{c} : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^u \to \mathbb{R}^p$ such that

$p << n + m + u$. Note that such a mapping always exists for $p \leq n + m + u$, since a trivial exact mapping is possible at $\dim(\boldsymbol{\phi}) = \dim(\mathbf{x}^*)$. Once a mapping is identified, the dynamics of the *latent variables* $\boldsymbol{\phi}$ can be embedded in the scheduling problem. The resulting scheduling problem has a low intrinsic dimensionality, with dynamics evolving only in the (low-dimensional) latent-variable space $\mathbb{R}^p$:

$$\max_{\mathbf{y}^{sp}} \quad \int_{t=0}^{t=t_f} J(\mathbf{y}, t)dt \tag{16.6}$$

$$s.t. \quad \dot{\boldsymbol{\phi}} = \boldsymbol{f}^{\phi}(\boldsymbol{\phi}, \mathbf{y}^{sp}) \tag{16.7}$$

$$\mathbf{x}^* \equiv \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \\ \mathbf{u} \end{bmatrix} = \boldsymbol{c}^{inv}(\boldsymbol{\phi}) \tag{16.8}$$

$$l(\mathbf{x}^*, t) = 0 \tag{16.9}$$

$$\mathbf{x}^{*L} \leq \mathbf{x}^* \leq \mathbf{x}^{*U} \tag{16.10}$$

Assuming that the mapping $\boldsymbol{c} : \mathbf{x} \rightarrow \boldsymbol{\phi}$ and the inverse mapping $\boldsymbol{c}^{inv} : \boldsymbol{\phi} \rightarrow \mathbf{x}$ exist, the dimensionality of the dynamic constraint(s) is now $p = \dim(\boldsymbol{\phi})$. If the mappings $\boldsymbol{c}(\cdot)$, $\boldsymbol{c}^{inv}(\cdot)$ are exact, and the dynamics of the latent variables are represented accurately by $\boldsymbol{f}^{\phi}(\cdot)$, then (16.6)–(16.10) is identical to the original scheduling problem (12.6)–(12.11). Note that for $\boldsymbol{c}(\cdot)$, $\boldsymbol{c}^{inv}(\cdot)$ to be exact, or equivalently, $\boldsymbol{c}^{inv}(\boldsymbol{c}(\mathbf{x}^*)) = \mathbf{x}^*$, $\mathbf{x}^*$ can only contain p independent variables. The remaining variables must be (nonlinearly) correlated. In practical situations, process variables that feature path constraints of the type in (16.10) may only be a subset of the full vector of process variables, and manifold learning can be carried out in a space of already lower dimension. Nevertheless, recent work [186, 283] has highlighted tradeoffs between dynamic production schedules and equipment fatigue, suggesting that some variables without explicit constraints may still be relevant in the scheduling layer and should be included in $\mathbf{x}^*$.

If the dynamics of $\mathbf{x}^*$ present a low-dimensional manifold only in a limit case (e.g., when the process dynamics are in a singularly perturbed form), the latent variables $\boldsymbol{\phi}$ only

approximate the true system. Specifically, $\boldsymbol{c}^{inv}(\boldsymbol{c}(\mathbf{x}^*)) \approx \mathbf{x}^*$, and the mappings $\boldsymbol{c}(\cdot)$, $\boldsymbol{c}^{inv}(\cdot)$ are inexact. In this case, some information is lost by "collapsing" the dynamics of $\mathbf{x}^*$ onto a reduced dimension, resulting in an approximation, $\mathbf{x}^{*\prime} = \boldsymbol{c}^{inv}(\boldsymbol{c}(\mathbf{x}^*))$. The accuracy of the approximation, in terms of $\| \mathbf{x}^* - \mathbf{x}^{*\prime} \|$, can be improved by increasing p until the original system is fully recovered at $p = n+m+u$ (p can be smaller if some variables are correlated). In other words, the dimension of the latent manifold p can be used as a parameter for adjusting the accuracy of the reduced-order representation of the closed-loop dynamics.

Remark 16.2. Reducing the number of dynamic variables in the original SBM problem (16.1)–(16.4) to the lower-dimensional problem (16.6)–(16.10) may be beneficial for both sequential [196] and simultaneous [129] dynamic optimization approaches. This chapter focuses on sequential approaches, where the Jacobian size for computing implicit time integration steps is reduced by limiting the number of dynamic variables to p. I expect the benefits to also extend to simultaneous approaches, where the number of differential state variables treated by the optimization problem is still reduced. However, note that the explicit dimensionality of $\mathbf{x}^*$, $\dim(\mathbf{x}^*)$, may remain larger than $\dim(\mathbf{w})$ in (16.1)–(16.4). The interested reader is referred to [268, 269] for an overview of sequential techniques for dynamic optimization and [28] for information on simultaneous strategies.

16.2.1 Latent Variable Scheduling Framework

The proposed approach for latent variable scheduling comprises the following steps:

(i) Obtain historical process operating data representative of typical production schedules

(ii) Learn latent variable mappings $\boldsymbol{c} : \mathbf{x}^* \rightarrow \boldsymbol{\phi}$ and $\boldsymbol{c}^{inv} : \boldsymbol{\phi} \rightarrow \mathbf{x}^{*\prime}$

(iii) Transform historical data $\mathbf{x}^*(t)$ using $\boldsymbol{c}$ to produce $\boldsymbol{\phi}(t)$

(iv) Determine model form and fit a dynamic model $\dot{\boldsymbol{\phi}} = \boldsymbol{f}^{\phi}(\boldsymbol{\phi}, \mathbf{y}^{sp})$ to the latent variables using the transformed data set

(v) Solve the low-dimensional scheduling problem (16.6)–(16.10) with path constraints on $\mathbf{x}^{*\prime} = \boldsymbol{c}^{inv}(\boldsymbol{\phi})$

16.2.2 Learning Latent Variables with Autoencoders

Autoencoders (AEs) provide a straightforward means for manifold learning, since they can simultaneously learn an explicit and complex (nonlinear) mapping $\boldsymbol{c}(\mathbf{x}^*)$ and an associated inverse mapping $\boldsymbol{c}^{inv}(\boldsymbol{\phi})$ using simple basis functions. Linear autoencoders operate in the same space as PCA [98], while nonlinear autoencoders can be viewed as a form of nonlinear PCA [141]. A brief overview of AEs as relevant to the current work is presented here; Chapter 14 of the book by Goodfellow et al. [98] provides a more complete discussion of autoencoders, their uses, and comparisons to other manifold learning techniques.

Briefly speaking, an *autoencoder* is a feed-forward artificial neural network that aims to replicate its input at its output. At a particular hidden layer within the autoencoder, the input is described as a "code," or $\boldsymbol{\phi}$. The dimensionality of the code $\boldsymbol{\phi}$ is determined by the structure of the neural network. The full network represents $(\boldsymbol{c}^{inv} \circ \boldsymbol{c})(\mathbf{x}^*)$. The autoencoder is naturally split into the layers leading to $\boldsymbol{\phi}$, or $\boldsymbol{\phi} = \boldsymbol{c}(\mathbf{x}^*)$, and the subsequent layers, or $\mathbf{x}^{*\prime} = \boldsymbol{c}^{inv}(\boldsymbol{\phi})$. The output of the network, $\mathbf{x}^{*\prime} = \boldsymbol{c}^{inv}(\boldsymbol{c}(\mathbf{x}^*))$ is an estimate of the original input $\mathbf{x}^*$. The autoencoder is typically trained using an iterative method by minimizing a loss function L:

$$L(\mathbf{x}^*, \mathbf{x}^{*\prime} = \boldsymbol{c}^{inv}(\boldsymbol{c}(\mathbf{x}^*))) \tag{16.11}$$

which penalizes discrepancies between $\mathbf{x}^{*\prime}$ and $\mathbf{x}^*$. Commonly used loss functions include the mean squared error, mean absolute error, and variations of the hinge and cross-entropy

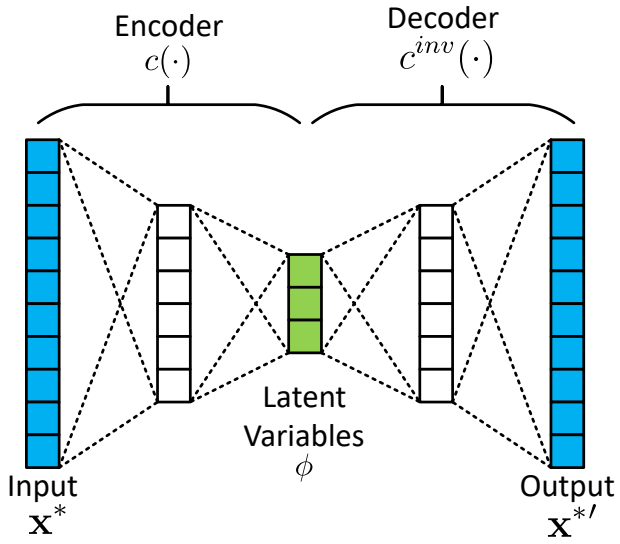

Figure 16.1: Conceptual depiction of an undercomplete autoencoder.

functions. For the purpose of learning a low-dimensional manifold underlying a set of input data, I employ *undercomplete autoencoders*, or those with $\boldsymbol{\phi}$ constrained to have a lower dimension than $\mathbf{x}^*$. By restricting, or "bottlenecking," information flow through the feedforward neural network, undercomplete autoencoders capture the salient trends present in the training data. Figure 16.1 depicts the structure of an undercomplete autoencoder with a two-layer encoder, two-layer decoder, and an encoded dimensionality of three.

Undercomplete autoencoders are often constructed with an encoder and decoder that each comprise a single hidden layer. The universal approximator theorem [115] guarantees that a feedforward neural network with at least one hidden layer can approximate any function (within a broad class) to an arbitrary degree of accuracy, given that enough hidden units are present. In practice, however, autoencoders with multiple hidden layers (termed *deep* autoencoders) can sometimes reduce the computational cost of representing certain functions, improve data compression, and/or decrease the amount of training data required [98, 114]. Thus, with enough hidden units through depth or breadth, any (nonlinear) map-

ping between $\mathbf{y} = \boldsymbol{h}(\mathbf{x})$ and $\mathbf{u} = \boldsymbol{K}(\mathbf{y}^{sp} - \boldsymbol{h}(\mathbf{x}))$ as relevant to the dynamical system under consideration here can be modeled to arbitrary accuracy, provided that $\boldsymbol{K}(\cdot)$ is bounded and continuous. MPC may not always satisfy this property, and alternative manifold learning techniques may be more suitable for systems exhibiting several distinct operating regimes.

16.2.3 Building Latent Variable Scale-Bridging Models

After a latent manifold underlying the closed-loop dynamics of a process is learned, the process operating data can be projected to the latent manifold to produce a low-dimensional representation. In particular, each observation $\mathbf{x}^*(t)$ can be transformed to $\boldsymbol{\phi}(t) = \boldsymbol{c}(\mathbf{x}^*(t))$. Then, given a data set (e.g., the data set used for manifold learning) of transformed observations, $\boldsymbol{\phi}(t)$, and process setpoints, $\mathbf{y}^{sp}(t)$, system identification can be performed in the latent variable space to create a (scale-bridging) model of the latent variable dynamics, $\dot{\boldsymbol{\phi}} = f^{\phi}(\boldsymbol{\phi}, \mathbf{y}^{sp})$. The system identification step can introduce additional inaccuracy in the dynamics embedded in the latent variable scheduling problem (16.6)–(16.10); however, this is equally true when identifying SBMs using physical process variables, as in Chapter 15. The identification of accurate dynamic models is a crucial step in both data-driven approaches and is performed using the same methods in either case.

Chapter 15 employed SBMs in the Hammerstein-Wiener (HW) form to capture the closed-loop process dynamics of actual, physical process variables for scheduling applications. This choice was motivated by the inherent structure of HW models. In contrast to unstructured dynamic models (e.g., recurrent neural networks [98]), HW models have fewer parameters and may be trained with significantly lower amounts of data. This is an important feature, since system identification experiments carried out on chemical plants can be expensive and time-consuming. Since HW models displayed adequate accuracies, I employ HW models to model the dynamics of latent variables in this chapter. The selection of

HW models for this study also facilitates comparison of system identification and scheduling results to related works [67, 196, 195, 258] involving data-driven models in the HW form.

16.3 Case Study: Demand-Response Scheduling of an Air Separation Unit

In this case study, I consider the single-product ASU as shown in Figure 16.2. The detailed mathematical model of the process dynamics is based on the work of Cao et al. [44], and is presented in full by Johansson [121]. Pattison et al. [196] investigated the scheduling problem using a full-order, detailed process model of the type (12.6)–(12.11), as well as the SBM scheduling problem (16.1)–(16.4). Dias et al. [67] developed a MPC system for the process and applied a novel simulation-optimization framework for integrated scheduling and control including MPC. I employ the MPC system and its associated state-space models given by Dias et al. [67] with slight modification. The mathematical model of the process and its control system are summarized below.

The process in Figure 16.2 produces high-purity nitrogen from an inlet air feed stream. The feed stream is compressed from atmospheric pressure to 6.8 bar, cooled, and passed through a primary (multi-stream) heat exchanger (PHX) where it is condensed against warming cryogenic streams. A portion of the air is removed from the PHX at an intermediate point and is sent to a turbine to generate electricity; the remainder exits the PHX at its saturation point. The two streams are combined and sent to the bottom of a cryogenic distillation column, which separates nitrogen from the other components of air. The bottoms product of the column is expanded through a valve before entering the reboiler. The reboiler and condenser are integrated in a single unit, allowing the bottoms stream to provide cooling duty to the condenser. The distillate of the column comprises the desired high-purity nitrogen stream. A portion of the distillate is sent to the condenser and becomes

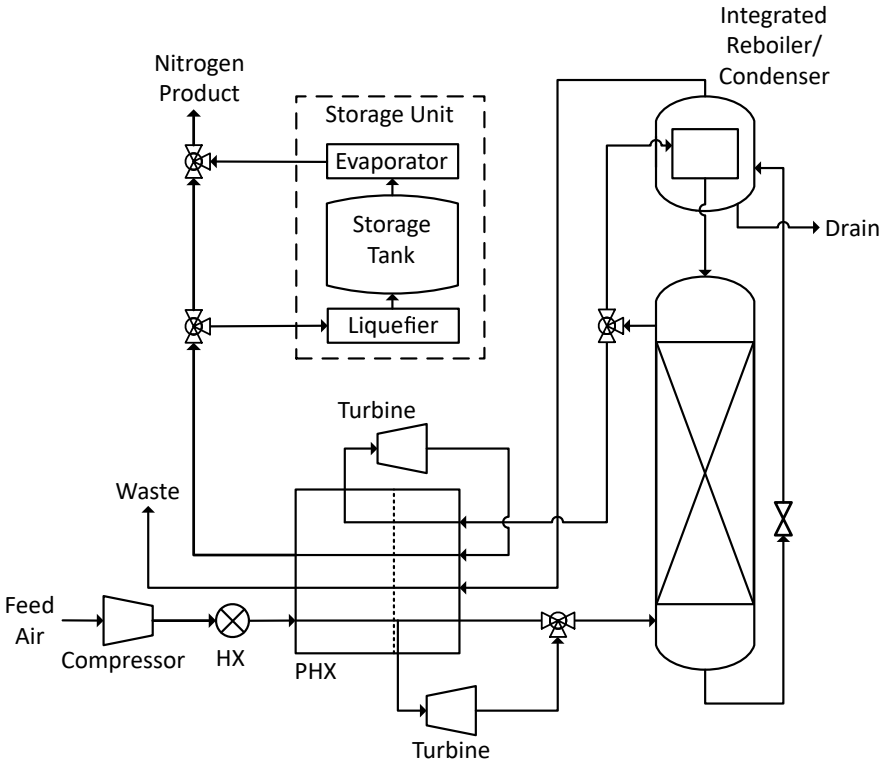

Figure 16.2: Diagram of a small nitrogen-production ASU with liquefier and liquid storage capacity.

the column reflux, while the remaining product stream is expanded in a second turbine after being vaporized in the PHX. The product stream and the waste nitrogen stream from the reboiler both pass through the PHX to provide cooling duty to the incoming air. The nitrogen liquefier, storage, and evaporator units are included in the flowsheet. These units allow the plant to liquefy and store excess gaseous nitrogen generated during periods of over-production, and conversely evaporate stored liquid nitrogen to satisfy gas nitrogen demand during times of under-production.

The full-order process model comprises 6,094 equations and has 430 differential variables. The entire model is implemented in gPROMS, and implementation details can be found in previous works [67, 196]. The ASU process is assumed to operate with a constant

gas nitrogen demand of 20 mol/s, with less than 2000 ppm impurity content (oxygen and argon). The process is assumed to be able to modulate its production rate by ±20% from its nominal value, representing a production range of 16 mol/s to 24 mol/s. The MPC for the process has four controlled variables and three manipulated variables. While the liquid drain stream from the reboiler was previously used as a fourth manipulated variable [67], I found that outputs are not sensitive to this input in the desired range of operation. The MPC variables are summarized in Table 16.1. The MPC has a sample time of six minutes and employs a linear state-space model created from system identification tests on the full-order dynamic model. The production rate setpoint represents $\mathbf{y}^{sp}(t)$, and its profile is set by the solution to the scheduling problem. The setpoints for the remaining controlled variables are fixed at $I_p^{sp} = 500$ ppm, ΔT_{IRC}^{sp} 2.2 K, and $M_{reb}^{sp} = 100$ kmol.

Table 16.1: Summary of MPC variables for the ASU process.

Controlled Variable (y)	Manipulated Variable (u)
Production flow rate	Inlet air flow rate
Product impurity	PHX split fraction
IRC temperature difference	Vapor product split
Reboiler liquid level	

16.3.1 Simulation Strategy for Generating Training and Testing Data

The augmented state variable vector for scheduling the ASU comprises 15 variables, i.e., $\dim(\mathbf{x}^*) = 15$: the seven variables of the MPC (Table 16.1), the power consumption, and seven state variables that feature constraints. These are the storage level M_{store}, column weeping ratio, column flooding ratio, column sump level, bubble-point pressure ratio, dew-point pressure ratio, nitrogen pressure ratio.

The column weeping ratio is defined as the minimal stage-wise ratio of vapor velocity to weeping velocity, while the flooding ratio is defined as the maximum stage-wise ratio of vapor velocity to flooding velocity. The bubble-point pressure ratio, defined as the ratio of

pressure to bubble-point pressure for the stream exiting the PHX must be greater than one to ensure the stream is fully liquefied. The dew-point and nitrogen point pressure ratios, defined as the ratio of pressure to dew-point pressure for, respectively, the feed stream drawn at an intermediate location in the PHX and the product stream passing through the turbine, must be less than one to ensure the streams are fully vapor-phase. Note that $\mathbf{x}^*$ contains several each of input variables, state variables, and output variables. The manipulated variables are included in $\mathbf{x}^*$ to understand the degree to which modulating plant operations is possible and potential effects on the equipment (e.g., see [283]).

A data set was simulated for manifold learning using the detailed first-principles process model described by Johansson [121]. The MPC was implemented "online" by linking the full-order dynamic model with the Model Predictive Control Toolbox in MATLAB [248]. The full-order model was run between each MPC interval to generate sampled state variable values, and the MPC problem was solved in MATLAB to provide updated setpoints for the local regulatory controller in the subsequent interval. To generate an operating data set that reflects production modulation, the SBM-based scheduling problem (16.1)–(16.4) was solved using electricity price data from a regional independent service operator (ISO). Ten two-day price signals were selected, aiming to include a wide gamut of prices and hence process closed-loop behaviors. In total, 20 days of operating data were included in the data set. The electricity prices and resulting production targets used to generate the data set are shown in Figure 16.3. Static dimensionality reduction techniques were applied to the simulated data set, and their statistics, as presented below, were computed using 5-fold cross-validation.

Principal component analysis (PCA) reveals the linear relationships and correlations present in $\mathbf{x}^*$ in the data set. The percentage of variance explained by each principal component for the full data set is shown in Figure 16.4. To ensure the correlation indicated by PCA is not coincidental, the same analysis was applied to the case where $\mathbf{x}^*$ included

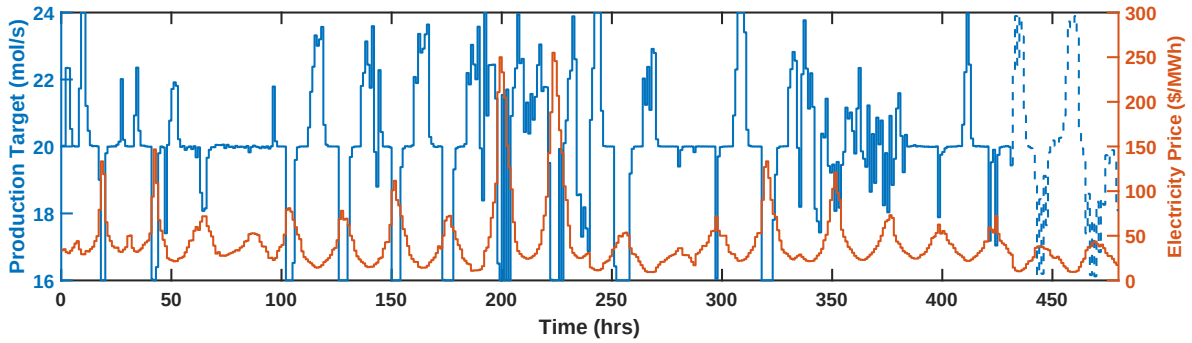

Figure 16.3: Electricity prices and production setpoints used to generate data set. The dashed lines depict the last 10% of the data, which are used for dynamic model validation.

all *process-level* variables. Here, the set of process-level variables refers to the properties (i.e., temperature, pressure, composition) of inter-unit streams and the operating conditions of the process units. Excluding variables that are necessarily identical to others and those with fixed/set values, there are 70 process-wide variables in total. Note that while the full dynamic model includes 6,094 equations, this analysis was limited to process-level variables, which provide ample information for most scheduling calculations. For example, many of the variables present in the model are associated with the spatial discretization of the primary heat exchanger and would not be measured in real-time in the plant.

PCA of the full data set produced a similar result to the case of 15 variables. In both cases, the percentage of variance explained decreases quickly with an increasing number of principal components (note the logarithmic ordinate scaling in Figure 16.4), suggesting that the closed-loop process dynamics of $\mathbf{x}^*$ can be "collapsed" to a lower dimension, and that the accuracy of the approximation can be tuned by carefully selecting the dimensionality of the reduced-order representation. The variance captured decays more slowly after approximately 12 components for the case of all process-wide variables, since only the first 15 components (of 70) are shown.

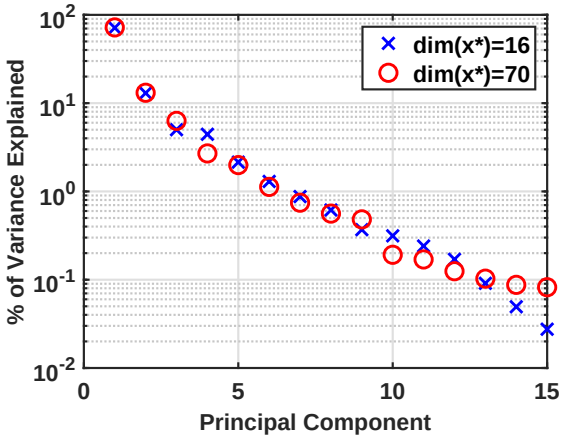

Figure 16.4: Percentage of variance in the full data set explained by each principal component.

16.3.2 Manifold Learning Results

Though PCA showed that the data set could be approximated reasonably using a low-dimensional set of linear latent variables, I next investigate nonlinear manifold learning techniques to further capture the closed-loop process dynamics. Manifold learning on the data set was performed using autoencoders (AEs). Several AE architectures were tested: *Tanh(2x)*, having `tanh` activation functions and one hidden layer in the encoder and decoder; *Tanh(3x)*, having `tanh` activation functions with two hidden layers in the encoder and decoder; and *Linear*, with linear activation functions. As mentioned above, the representation power of an AE can be increased through increasing the depth or breadth of the neural network. In this chapter, the breadth of hidden layers is fixed to the truncated average of the input dimension and the encoded dimension. The effect of increasing the network size is investigated by switching from a single hidden layer to two hidden layers.

Note that adding hidden layers to a neural network with only linear transformations does not increase the representation power of the model, since linear combinations of linear basis functions remain linear.

Each process variable was scaled to take values between 0-100%, with 0% representing its minimum value in the data set and 100% representing its maximum. The AEs were implemented and trained using TensorFlow [1] with the Adam solver [133] and the mean squared error (MSE) as the loss function:

$$\text{MSE} = \frac{||x_{ref} - x||^2}{N_s} \tag{16.12}$$

where N_s is the number of samples. Each AE was trained until the validation loss function remained the same or increased for several straight epochs. The cross-validated test MSEs from training the AEs are shown in Figure 16.5. While errors in the predictions for each process variable had the same weights in the loss function for this study, the errors of individual variables could be weighted differently to prioritize accuracy in certain process variables.

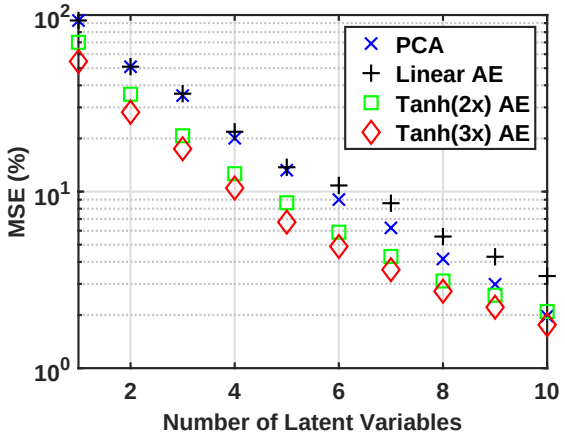

Figure 16.5: Comparison of validation MSE from dimensionality reduction techniques on process variables. The plotted MSEs were computed with 5-fold cross validation.

The results confirm the observation from the initial PCA, that the closed-loop process dynamics can be "collapsed" to a lower dimension. For example, with ten dimensions, the tested methods can represent the complete input data very well, with MSEs around 2%.

PCA presents an adequate manifold learning baseline, and linear AEs operate in the same space [98]; however, I found that the accuracy of linear AEs to be lower than PCA when the number of latent variables included was large. This deviation can be attributed to difficulty in training a large AE using a stochastic optimization procedure, while PCA computes the optimal solution analytically. The nonlinear AEs achieve lower MSEs than the linear AE and PCA in all cases, with the benefits being significant especially at lower manifold dimensionality. The nonlinear AEs are capable of learning more complex relationships present in the data [141], and the increase from one hidden layer to two layers further increases representation power. The benefit of nonlinear mappings diminishes as the number of latent variables increases.

Effect of Measurement Noise. Dimensionality-reduction techniques are often employed for their ability to filter noisy data. Noise may sometimes be artificially introduced during autoencoder training to improve generalization ability [98]. By constraining the intrinsic dimensionality of the retained information, latent variables retain the most important dynamics. In this case study, measurement noise was simulated by adding 5% normally distributed error to all process variables in the training data set. AEs with the same configurations as described above were trained, and the MSEs are shown in Figure 16.6. Similar to the cases without measurement noise (Figure 16.5), the nonlinear AEs provide better process representations when the desired dimensionality is low, and the benefits of a nonlinear model decrease as more latent variables are added. All the models are less accurate when measurement noise is added, though this accuracy could be improved by increasing the size of the data set.

Insight into the denoising ability of the learned models can be obtained from examining their accuracy in predicting the original data set (i.e., the "ground truth" data without measurement noise). The MSEs of the models shown in Figure 16.6 evaluated against the

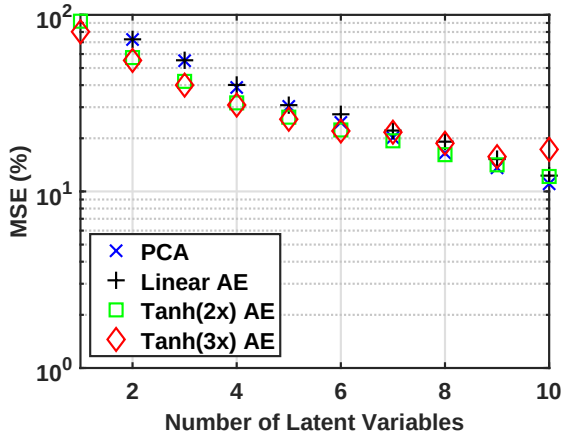

Figure 16.6: Comparison of validation MSE from dimensionality reduction techniques with 5% normally distributed measurement noise. The plotted MSEs were computed with 5-fold cross validation.

ground truth data are shown in Figure 16.7. The MSEs decrease rapidly as the number of latent variables increases from one to six, where the model accuracy plateaus at approximately 11% MSE. Interestingly, the learned models are generally more accurate in predicting the ground truth data than the noisy data (Figure 16.6), demonstrating their ability to filter noise. The introduction of measurement noise can be likened to a form of regularization, where shifting the variance-bias tradeoff improves model generalizability at the cost of some accuracy (the MSEs in Figure 16.7 are higher than those in Figure 16.5). Since least-squares regression filters progressively more Gaussian noise as the number of samples increases, I expect that the model accuracies could be improved by increasing the size of the data set.

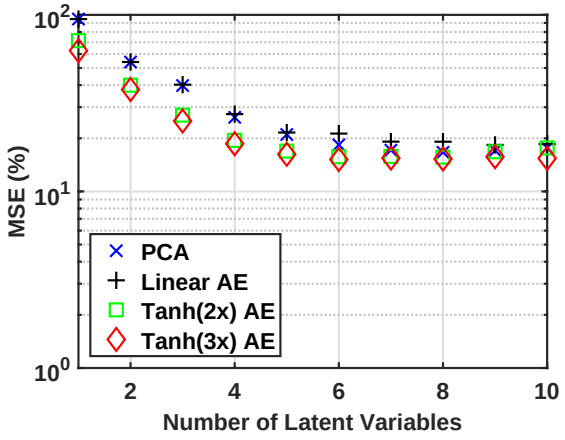

Figure 16.7: Comparison of validation MSE computed against "ground truth" from dimensionality reduction techniques with 5% normally distributed measurement noise. The plotted MSEs were computed with 5-fold cross validation.

Effect of Additional Process Variables. To confirm that the low-order manifold mapping for $\mathbf{x}^*$ is not enabled by coincidentally selecting 15 correlated variables, the same AEs were trained on the full $\mathbf{x}^*$ vector that includes all 70 *process-level* variables, as described in Section 16.3.1. The cross-validated test MSEs from training the AEs on the full vector of process-wide variables are shown in Figure 16.8. The dimension of the hidden layers in the nonlinear AEs was again chosen to be the truncated average between the input layer and latent variable dimensions. The AEs were implemented and trained using the same procedure as above. Note that the AEs have more units (and representation power) due to the increase in dimension of the input layer.

As expected, increasing the dimension of $\mathbf{x}^*$ does not have a significant effect on the manifold learning procedure, since the intrinsic dimension $\dim_i(\mathbf{x}^*)$ remains unchanged. This result supports the assertion that the dimensionality reduction is enabled by the low intrinsic dimension of the system. I again find that the closed-loop process dynamics of all 70 variables can be "collapsed" to a low dimension, with each additional dimension having

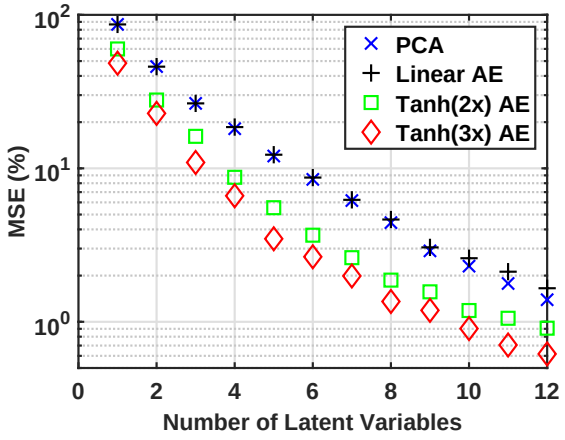

Figure 16.8: Comparison of validation MSE from dimensionality reduction techniques on all 70 process-level variables. The plotted MSEs were computed with 5-fold cross validation.

a diminishing impact on model accuracy. The linear AE again exhibits a similar result to performing PCA on the data set, while the nonlinear AEs again perform better than both linear methods. The nonlinear AEs show improved accuracy compared to Figure 16.5 since they have more hidden units. The MSEs for the nonlinear AEs reach $\sim$1% with ten latent variables, while the linear models reach MSEs of around 2%.

Remark 16.3. Effectiveness of the manifold learning procedure was further verified using the industrial, real-world dataset described in Chapter 15. The cross-validated test MSEs from training the AEs on the full vector of 143 variables are shown in Figure 16.9. The results suggest the increased importance of *nonlinear* manifold learning for real-world processes. Furthermore the MSEs of the nonlinear models plateau after $\sim$15 latent variables are used, suggesting the likely presence of measurement noise, as in Figure 16.7.

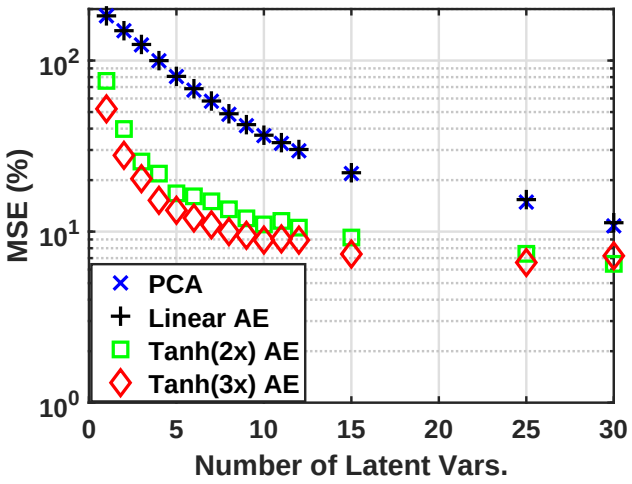

Figure 16.9: Comparison of validation MSE from dimensionality reduction techniques on 143 process variables from the industrial ASU described in Chapter 15. The plotted MSEs were computed with 5-fold cross validation.

16.3.3 Dynamic Modeling Results

Given the low MSEs possible with increasing p, I expect system identification to be the limiting factor in model accuracy for this study. Pattison et al. [196] found that a 10% "back-off" constraint was needed to compute feasible schedules for the ASU with HW models of physical variables, providing insight into SBM accuracy. Therefore I select two low-dimensional representations of the ASU process dynamics with approximately 10% MSE (Figure 16.5): a linear AE with six latent variables and a nonlinear AE with one hidden layer and five latent variables. To investigate the effect of adjusting p, I also test two representations with approximately 20% MSE: a linear AE with four latent variables and a nonlinear AE with one hidden layer and three latent variables. For dynamic system identification, the first 18 days were used as training data (90% of the data set). The process variables $\mathbf{x}^*$ were encoded using the respective encoders to give $\boldsymbol{\phi} = \boldsymbol{c}(\mathbf{x}^*); \boldsymbol{\phi} \in \mathbb{R}^p$. The remaining two days are shown as dashed lines in Figure 16.3 and were left as test data.

The test data were generated using electricity price data from a month not included in the training data to account for the potential for new patterns to emerge in the production schedules. The effect of dimensionality reduction on the test data was evaluated as a baseline by computing their estimated values $\mathbf{x}^{*\prime} = \boldsymbol{c}^{inv}(\boldsymbol{\phi})$ using the true values of $\boldsymbol{\phi} = \boldsymbol{c}(\mathbf{x}^*)$. The profiles of the variables $\mathbf{x}^*$ in the test data set, as well as their estimated values using all four AEs, are shown in Figure 16.10. Scatter plots of $\mathbf{x}^{*\prime}$ against $\mathbf{x}^*$ are shown in Figure 16.11 for the most inaccurately predicted variables with the nonlinear AEs, where the improvement in prediction accuracy from increasing p can easily be seen.

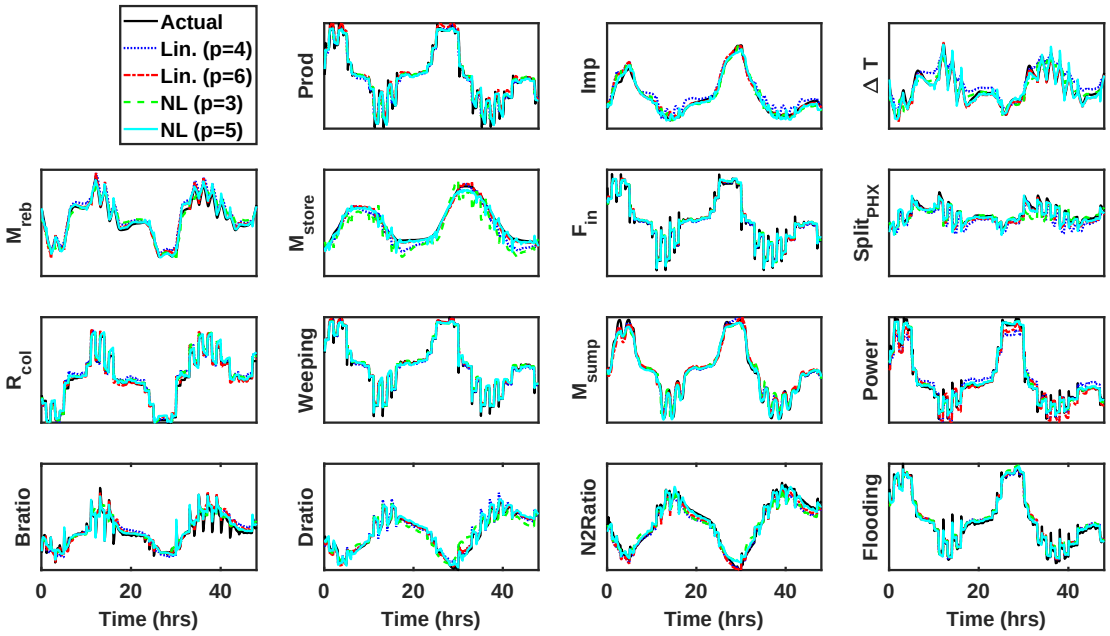

Figure 16.10: Evolution of the process variables predicted using various reduced-order representations, given "true" values of the latent variables.

Hammerstein-Wiener (HW) models were fitted to the dynamics of the *latent* variables $\dot{\phi}_i = f_i^{\phi}(\phi_i, \mathbf{y}^{sp})$. The models were fitted using the System Identification Toolbox in MATLAB [248]. Piecewise-linear and polynomial transformations were used to model the Φ and Ψ blocks of (15.1)–(15.4). The form and polynomial order/number of piecewise segments

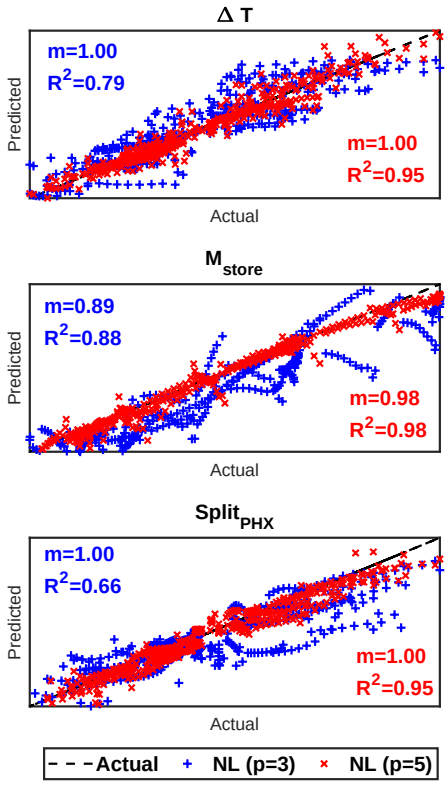

Figure 16.11: Scatter plots of a few process variable values predicted by nonlinear (NL) autoencoders with $p = 3$ and $p = 5$ latent variables. The predictions, $\mathbf{x}^{*\prime} = (c^{inv} \circ c)(\mathbf{x}^*)$, are plotted against their true values, $\mathbf{x}^*$, in the test data set.

for each transformation was determined by minimizing the normalized Akaike information criterion (nAIC) while using a large number of piecewise-linear segments for the other transformations and a high-order linear state-space model. The order of each linear state-space model was similarly determined using the nAIC. The resulting HW model structures and normalized mean squared error (NMSE) are shown in Tables 16.2 and 16.3. Note that a higher NMSE indicates a better fit, in contrast to MSE. The NMSE values are computed using the `goodnessOfFit()` function in MATLAB, which uses the following formula:

$$\text{NMSE} = 1 - \frac{||x_{ref} - x||^2}{||x_{ref} - \text{mean}(x_{ref})||^2} \tag{16.13}$$

Table 16.2: Hammerstein-Wiener model structures and accuracies for linear latent variables. Nonlinear transformations are denoted with 'pwl' for piecewise-linear and 'poly' for polynomial.

	Linear ($p = 4$)				Linear ($p = 6$)			
	Φ		Ψ	NMSE	Φ		Ψ	NMSE
Variable	Form	n_d	Form	train/test	Form	n_d	Form	train/test
ϕ_1	pwl-1	4	poly-2	0.77/0.78	pwl-3	5	pwl-2	0.83/0.86
ϕ_2	pwl-2	4	pwl-1	0.78/0.88	pwl-1	6	pwl-3	0.72/0.74
ϕ_3	pwl-2	5	pwl-1	0.93/0.96	pwl-1	4	poly-2	0.90/0.93
ϕ_4	pwl-2	5	poly-2	0.54/0.51	pwl-4	4	pwl-4	0.70/0.79
ϕ_5	-	-	-	-	poly-2	4	pwl-3	0.85/0.90
ϕ_6	-	-	-	-	pwl-4	4	pwl-5	0.64/0.63
average	-	-	-	0.76/0.78	-	-	-	0.77/0.81

Table 16.3: Hammerstein-Wiener model structures and accuracies for nonlinear latent variables. Nonlinear transformations are denoted with 'pwl' for piecewise-linear and 'poly' for polynomial.

	Nonlinear ($p = 3$)				Nonlinear ($p = 5$)			
	Φ		Ψ	NMSE	Φ		Ψ	NMSE
Variable	Form	n_d	Form	train/test	Form	n_d	Form	train/test
ϕ_1	pwl-3	5	pwl-3	0.76/0.83	pwl-5	8	pwl-2	0.49/0.38
ϕ_2	pwl-4	4	pwl-2	0.77/0.89	poly-2	8	pwl-1	0.74/0.83
ϕ_3	pwl-3	6	poly-3	0.53/0.70	pwl-2	4	poly-3	0.89/0.90
ϕ_4	-	-	-	-	pwl-3	4	pwl-3	0.69/0.83
ϕ_5	-	-	-	-	pwl-2	5	pwl-2	0.70/0.77
average	-	-	-	0.69/0.81	-	-	-	0.70/0.74

The latent variable HW models were then simulated with the respective decoder $\boldsymbol{c}^{inv}(\cdot)$ (created during autoencoder training) incorporated as additional static equalities. The simulations provide estimates of the latent variables $\boldsymbol{\phi}$ and decoded estimates of the process variables $\mathbf{x}^{*\prime} = \boldsymbol{c}^{inv}(\boldsymbol{\phi})$. The actual variable profiles $\mathbf{x}^*$ and the estimated process variable profiles $\mathbf{x}^{*\prime}(t)$ are shown in Figure 16.12. The NMSEs for all 15 variables are shown in Table 16.4. While the SBMs for the latent variables exhibited lower NMSEs (Tables 16.2 and 16.3), the final predictions for the process variables have NMSEs (Table 16.4) comparable to (or even higher than) previous works. Pattison et al. [196] reported an average validation NMSE of 83.75% using HW models to directly represent the behavior of eight physical variables.

The agreement between the actual values and the estimated profiles is generally very good, confirming the closed-loop process dynamics are approximated well by a data-driven model whose dynamics are confined to a low-dimensional, intrinsic manifold. As expected, increasing the dimensionality of the latent manifold from four to six in the linear case, and from three to five in the nonlinear case, improves the accuracy of the model predictions. The predictions of the integrated reboiler-condenser temperature difference ΔT_{IRC} and the PHX split fraction suffer from the largest inaccuracy. The predictions are slightly improved by increasing dimensionality (Figure 16.11), but a comparison between Figures 16.10 and 16.12 suggests that the error is primarily introduced by the system identification step. I note that the identified, data-driven models may suffer from overfitting, particularly in cases where the training/test data set does not include the current operational situation. Implementation may benefit from a monitoring technique to detect whether the plant is entering an operating regime that has not been previously explored, at which point models should be re-identified.

Table 16.4: NMSEs found on validation data set with linear (Lin) and nonlinear (NL) autoencoders at various levels of dimensionality reduction. Values below 0.75 are in bold.

	NMSE			
Variable	Lin ($p = 4$)	Lin ($p = 6$)	NL ($p = 3$)	NL($p = 5$)
Production	0.95	0.96	0.95	0.97
Impurity	0.92	0.91	0.81	0.93
ΔT_{IRC}	**0.63**	0.90	**0.68**	0.86
M_{reb}	0.83	0.89	0.86	0.94
M_{store}	0.86	0.85	**0.71**	0.90
Air Flow	0.98	0.98	0.98	0.97
PHX Split	**0.21**	**0.52**	**0.10**	**0.31**
R_{col}	0.96	0.95	0.94	0.95
Weeping Ratio	0.96	0.97	0.97	0.98
Sump Level	0.95	0.94	0.93	0.93
Power	0.91	0.94	0.97	0.98
B Ratio	**0.71**	0.88	0.80	0.77
D Ratio	0.89	0.91	**0.70**	0.86
N_2 Ratio	0.93	0.95	0.87	0.96
Flooding Ratio	0.98	0.97	0.97	0.98
Average	0.85	0.90	0.83	0.89

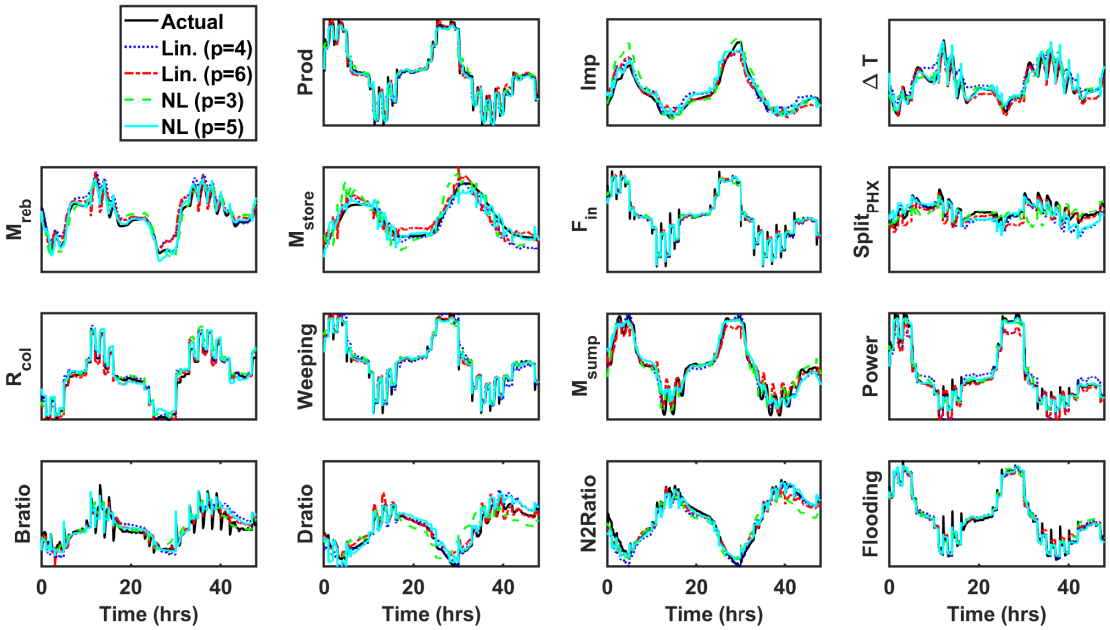

Figure 16.12: Evolution of the process variables predicted using various reduced-order representations, given values for the latent variables predicted by the identified HW models.

16.3.4 Optimal Scheduling Results

The latent-variable scheduling optimization problem (16.6)–(16.10) was solved for the DR operational scenario considered in [67]. In this scenario, the storage tank is assumed to have a maximum capacity of 200 kmol of liquid nitrogen, with an initial inventory of 50 kmol. The inventory must be returned to at least its initial value at the end of the scheduling horizon to avoid reporting false economic benefits derived from selling pre-existing inventory. The day-ahead electricity prices are assumed to be known over a 48-hour horizon, and there is a constant demand of 20 mol/s for the gas nitrogen product (equal to the nominal capacity of the plant). I consider here only the "offline" scheduling problem (with no re-scheduling); however, the proposed framework allows the scheduling problem to be solved quickly (∼100-200s), which may benefit online scheduling techniques in the future [23, 195].

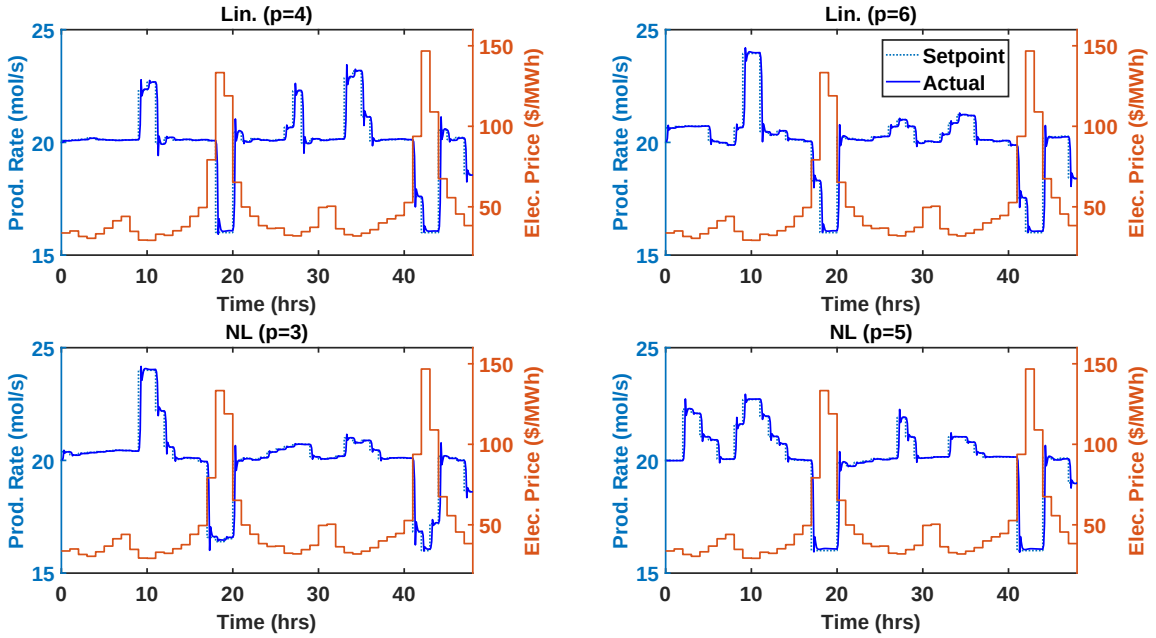

Figure 16.13: Electricity prices and the corresponding optimal production schedules.

Results Obtained Using Linear Latent Variable Models. The 48-hr scheduling problem was solved with the proposed latent-variable approach, using the linear mappings with $p = 4$ and $p = 6$. The models were implemented in gPROMS [204], and optimization was carried out using the built-in sequential dynamic optimization solver. The presented optimal points represent local optima found using 20 mol/s as the initial guess for the production setpoint at all times. The calculations were performed on a 64-bit Windows system with Intel Core i7-8700 CPU at 3.20 GHz and 16GB RAM. The implementation with four linear latent variables includes 29 differential variables and 64 total variables, while that with six linear latent variables includes 38 differential variables and 79 total variables. The scheduling problem with four latent variables was solved in 52 iterations, using 104.7s of CPU time (2.0s per iteration on average). The problem with six latent variables was solved in 42 iterations, using 113.0s of CPU time (2.7s per iteration on average). The two optimal

schedules were simulated using the full-order dynamic model. The number of iterations taken by the local optimization solver to solve each problem is unpredictable. Nevertheless, the time per iteration is fairly consistent within each problem, and the number of iterations can be limited for an expedited, but suboptimal, solution [48].

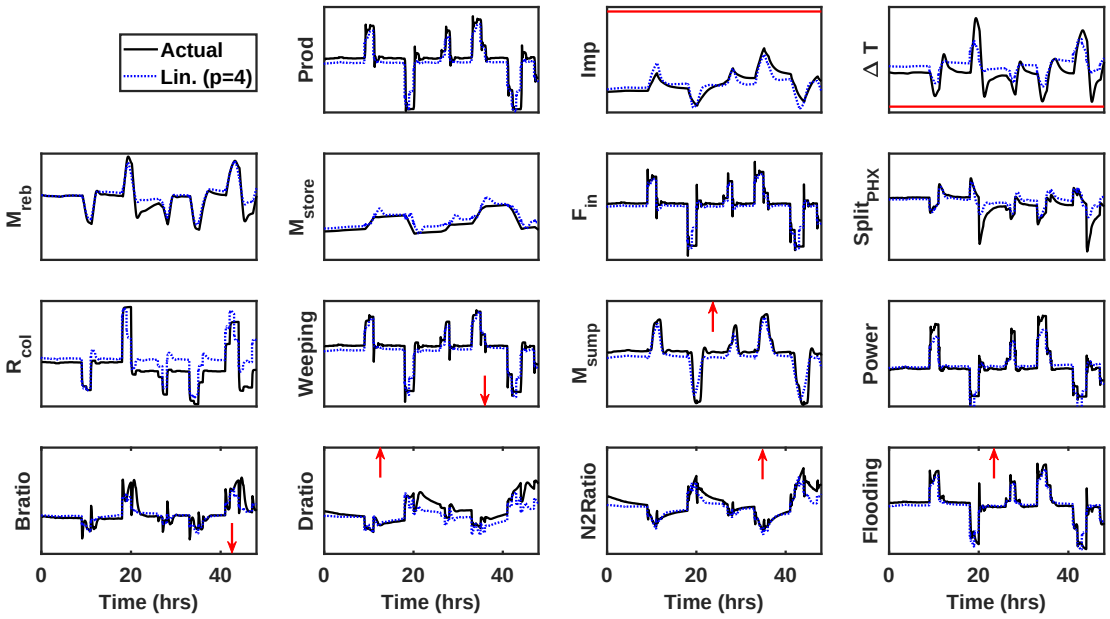

Figure 16.14: Optimal schedule generated with linear AEs (p=4). "Actual" profiles are generated by simulation of the same schedule using the full-order model with online MPC. Variable bounds are shown in red. Ordinate limits are 0-100% of the respective scaled variable. Some bounds/constraints do not appear within this scaling; their locations are indicated with an arrow (e.g., an arrow pointing upwards indicates that an upper bound exists).

The production rates found in simulation of the optimal schedules are shown in Figure 16.13, along with their setpoints/targets, which are closely tracked. As expected, production rates are scheduled to decrease when energy prices are high in both schedules. The behaviors of all 15 modeled process variables in the two computed schedules are shown in Figures 16.14 and 16.15. The temperature driving force across the reboiler/condenser nearly reaches its

bound in both schedules, but this potential constraint violation is only predicted by the low-dimensional representation when six linear latent variables are included. As shown in Table 16.4, increasing p from four to six improves the test NMSE on ΔT_{IRC} from 0.63 to 0.90. None of the other variable path constraints were reached when the optimal schedules were simulated with the full-order dynamic model. The end point constraints of returning the storage and reboiler levels to at least their initial conditions were also met in both schedules. Although the variables with inactive constraints may not have been necessary for computing a feasible schedule, the proposed approach captures the dynamics of all constrained variables in the scheduling problem. This eliminates the difficult task of anticipating which constraints may be violated and should therefore be modeled.

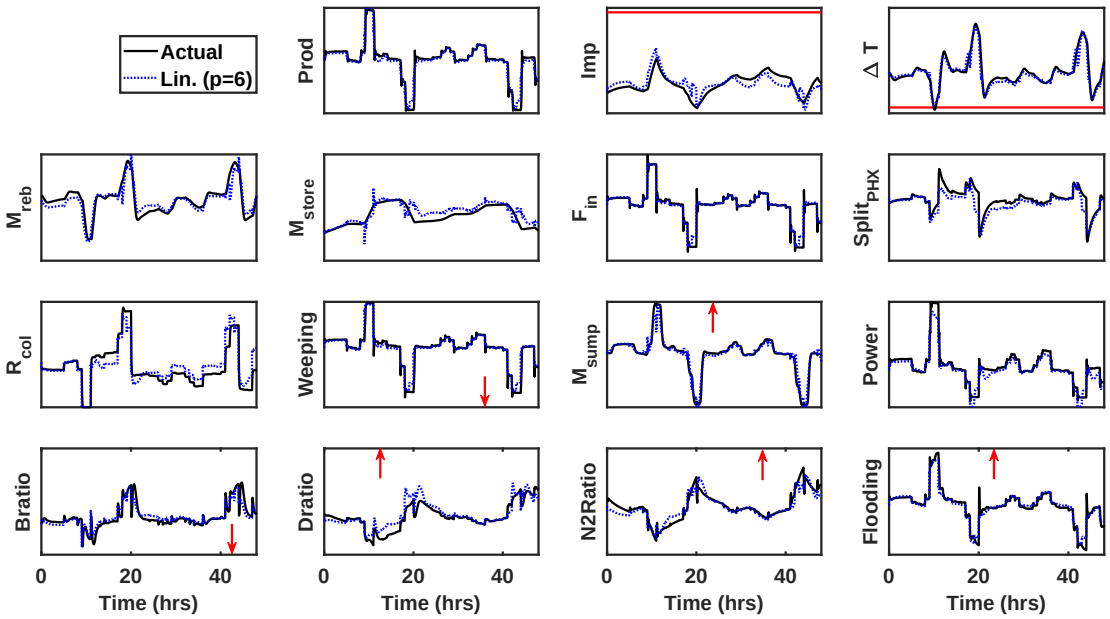

Figure 16.15: Optimal schedule generated with linear AEs (p=6). "Actual" profiles are generated by simulation of the same schedule using the full-order model with online MPC. Variable bounds are shown in red. Ordinate limits are 0-100% of the respective scaled variable. Some bounds/constraints do not appear within this scaling; their locations are indicated with an arrow.

The predictions are generally improved by including six (vs. four) latent variables, especially in the aforementioned temperature driving force ΔT and PHX split fraction. However, some deviations are visible for the column reflux R_{col} predictions with both models (Figures 16.14 and 16.15). The operational costs calculated using the full-order dynamic model of the computed schedules are shown in Table 16.5. Both schedules result in a approximately 1% savings compared to a constant production profile set at the nominal rate (subject to the same electricity price profile). These savings are similar to those reported in [67] and represent a substantial amount in the context of the well-established, commoditized air separation industry. In contrast to the previous approaches, the proposed method maintains the computational efficiency of scheduling in a reduced dimension while providing predictions of all constrained variables.

Table 16.5: Optimal schedule economic results.

Case	Cost	Difference from baseline	Solution time
Baseline[a]	\$707.91	0%	-
Simulation-optimization[b]	\$698.30	1.4%	381s
Physical SBMs[c]	\$698.60	1.3%	610s
Lin. ($p = 4$)	\$700.75	1.0%	104s
Lin. ($p = 6$)	\$699.89	1.1%	113s
NL ($p = 3$)	\$701.65	0.9%	193s
NL ($p = 5$)	\$700.09	1.1%	238s

[a] Baseline denotes the constant production rate case
[b] Optimal point found using a simulation-optimization framework by [67]. Only variables involved in the MPC were modeled and constrained during optimization.
[c] Optimal point found using SBMs identified for eight physical process variable, as reported by [67]. Details of the SBM models are provided by [196].

Results Obtained Using Nonlinear Latent Variable Models. The same 48-hr latent-variable scheduling problem was solved using the nonlinear mappings with $p = 3$ and $p = 5$. The same implementation and optimization settings were used, but the models include a hidden layer and nonlinear transformations. The implementation with three nonlinear latent variables includes 25 differential variables and 66 total variables, while that with five nonlinear latent variables includes 38 differential variables and 86 total variables. The scheduling problem with three nonlinear latent variables was solved in 69 iterations, using 192.8s of CPU time (2.8s per iteration on average). The problem with five latent variables was solved in 61 iterations, using 237.8s of CPU time (3.9s per iteration on average). Although the problems with four linear and three nonlinear latent variables have a similar number of variables, the optimization problem with nonlinear latent variables requires more time per optimization iteration. The same phenomenon is observed for the problems with six linear and five nonlinear latent variables. The slowdown can be attributed to the nonlinear `tanh` transformations limiting integration step sizes and thereby slowing down the implicit time-integration scheme at each iteration.

The two computed optimal schedules were simulated with the aforementioned MPC and the full-order dynamic model. The production rate setpoints and actual values found at the optimal points are again shown in Figure 16.13. The behavior of all 15 modeled process variables in the two computed schedules is shown in Figures 16.16 and 16.17. The temperature driving force across the reboiler/condenser slightly violates the respective bound in the schedule computed with three nonlinear latent variables, which is not predicted accurately by the reduced-order model. Increasing p from three to five improves the test NMSE on ΔT_{IRC} from 0.68 to 0.86 (Table 16.4), and the constraint violation is avoided by using five nonlinear latent variables. The end point constraints of returning the storage and reboiler levels to at least their initial conditions were met in both schedules.

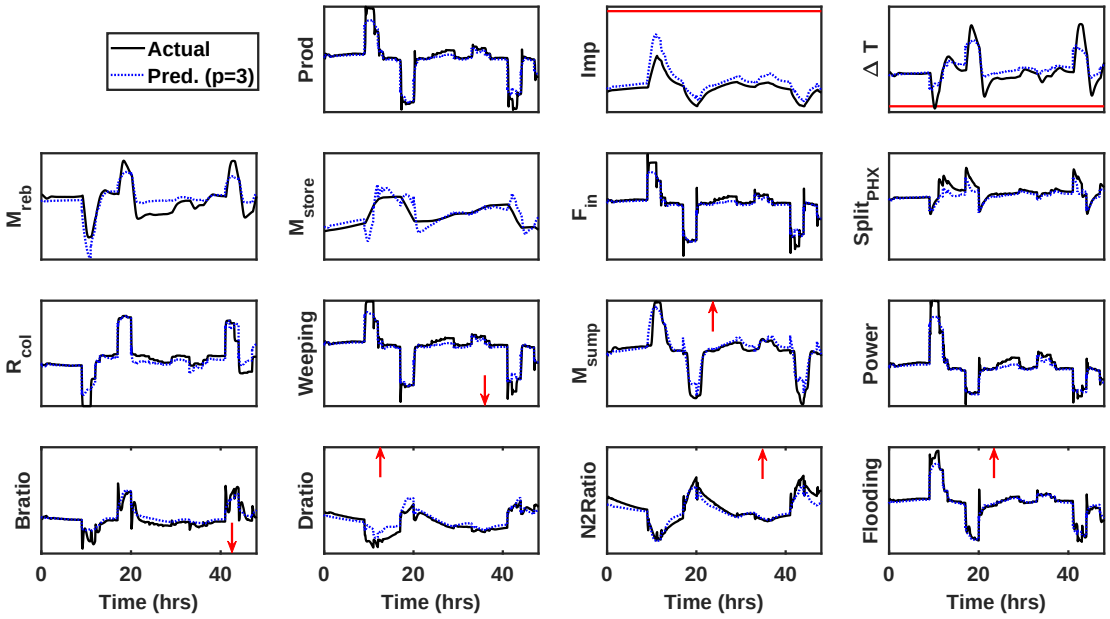

Figure 16.16: Optimal schedule generated with nonlinear AEs (p=3). "Actual" profiles are generated by simulation of the same schedule using the full-order model with online MPC. Variable bounds are shown in red. Ordinate limits are 0-100% of the respective scaled variable. Some bounds/constraints do not appear within this scaling; their locations are indicated with an arrow.

The optimization problem with dynamics represented by five nonlinear latent variables demonstrates the most accurate predictions of the evolution of process variables, but also required the most time per optimization iteration out of the four tested latent-variable scheduling problems. However, the optimization problem size is still greatly reduced from previous approaches [67, 195, 196], and correspondingly, the optimal schedule was obtained with less computational effort. In addition, the proposed formulation allows for more information on the process dynamics to be captured in scheduling calculations, with all constrained process variables predicted relatively accurately. The results in Section 16.3.2 further suggest that more process-level variables could be easily included at similar levels of accuracy without increasing the size of the latent dynamics. The intrinsic, low-dimensional

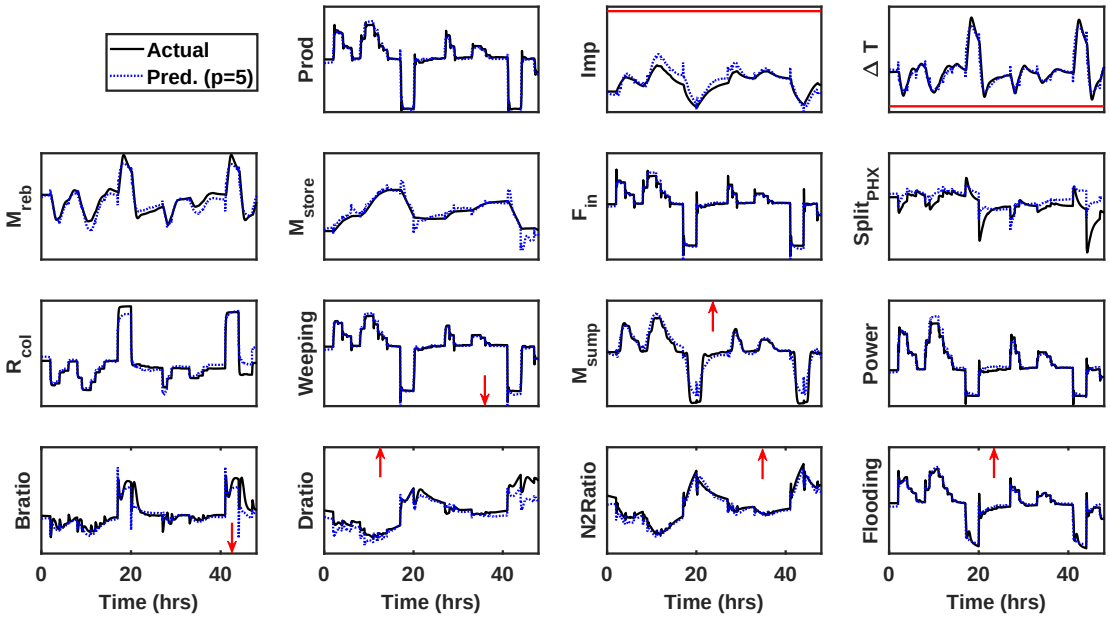

Figure 16.17: Optimal schedule generated with nonlinear AEs (p=5). "Actual" profiles are generated by simulation of the same schedule using the full-order model with online MPC. Variable bounds are shown in red. Ordinate limits are 0-100% of the respective scaled variable. Some bounds/constraints do not appear within this scaling; their locations are indicated with an arrow.

dynamics underlying the closed-loop system would be approximated, and the decoder could be expanded to include more process variables.

16.4 Summary

The integration of closed-loop process dynamics in production scheduling calculations is key to ensuring that production schedules do not violate process constraints when implemented in practice (i.e., the schedules are "dynamically feasible"). Driven by the need for computationally efficient representations of process dynamics, this chapter exploited the low intrinsic dimensionality of closed-loop process behavior to generate reduced-order models. I proposed a data-driven approach for learning a low-dimensional latent manifold underlying

variations in recorded observations, which can then be used to represent the process behavior. I presented a conceptual analysis of the existence of such a manifold, and I demonstrated the means for selecting the dimensionality of the latent manifold, so as to balance between complexity of the captured dynamics and model size. After projecting process variables to a latent manifold, system identification is only necessary for a lower number of (latent) variables, and the scheduling calculations require less computational expense. The framework was applied to the scheduling of an ASU, and the results confirm that the new approach retains more information about the process dynamics, compared to previous works, and simultaneously reduce the required computational effort.

Chapter 17

Conclusions and Future Work

This dissertation presented several new concepts which advance process design and operational optimization in the context of unconventional processing units and feedstocks. In this chapter, broad conclusions are drawn from each part of the dissertation, and future research directions are given.

17.1 Multi-Resolution Process Design Optimization

Part I of this dissertation presented concepts related to the optimization of steady-state process flowsheets involving detailed physical models of (some key) unit operations. Several examples were provided: a distributed model of a heat-quenched reactor, a rigorous model of a dividing-wall distillation column, a geometric model of a spiral-wound heat exchanger, and a rate-based model of a packed bed absorber. The models, and flowsheets incorporating these models, were simulated and optimized using pseudo-transient continuation as a robust numerical method, and the results demonstrated the need for and benefits of multiscale, multi-resolution modeling, simulation, and optimization for integrated and intensified processes. Importantly, these techniques bridge the gap between equipment and flowsheet design, which are typically performed in an iterative manner. Making these decisions in a single, unified step will improve the optimality of the resulting designs. Furthermore, I presented a technique for accelerating the convergence of pseudo-transient flowsheets, based on defining a hierarchy of time scales and sequentially enforcing quasi-steady-state conditions.

There are many potential avenues for future work in the area of multi-resolution flowsheet simulation and optimization (and pseudo-transient flowsheeting). It is firstly important to address the dynamic stability and convergence of pseudo-transient flowsheet models in the case where the original system of equations has multiple steady-state solutions (i.e., multiplicity). Furthermore, all of the optimization results presented were obtained using *local* optimization solvers, but pseudo-transient models can potentially be incorporated into *global* optimization strategies. For example, the models may be used to initialize flowsheets for reduced-space global optimization [34], to quickly solve upper-bounding problems, and/or to incorporate interval schemes originally introduced for dynamic optimization.

Finally, the pseudo-transient flowsheet methods could be significantly enhanced by the development of custom (stand-alone) software. The simulations and optimizations presented in this dissertation were primarily carried out in gPROMS [204], which has a reliable methods for the time integration of DAE system. However, I anticipate that better computational performance and ease of use could be obtained with a numerical solver that executes the quasi-steady-state schemes automatically, a built-in library of unit models, a custom modeling interface, etc. There are significant challenges involved here, most notably pertaining to the necessity for a physical properties package for most practical applications. Here, the pseudo-transient flowsheet approach could also be extended to incorporate models of molecular-level phenomena into flowsheet optimization, which could potentially replace (some) physical properties calculations [130].

17.2 Process Design for Variable Operation

In Part II of this dissertation, a novel framework was presented for solving process design optimization problems under uncertainty. Specifically, the design optimization problem was reformulated as a dynamic optimization, where the uncertain parameters are represented

as (continuous) dynamic disturbance variables in pseudo-time. The parameter uncertainty space can thus be explored efficiently with a continuous pseudo-time trajectory rather than via a finite number of samples as is the case in conventional scenario-based approaches for optimization under uncertainty. Moreover, it was demonstrated through numerous case studies that the proposed approach offers massive, order-of-magnitude reductions in computational effort (both CPU time and memory use), particularly in the case when the number of uncertain parameters and size of the optimization problem increase. These developments enable the use of more complex and accurate models during design under uncertainty, which in turn ensures that designs are truly feasible (i.e., implementable in practice).

Future research in this area should first investigate how the computational performance of this method is affected by parallel computing schemes (note that several strategies for parallelization were provided herein). It is also important to study how the performance using parallelization compares with comparable multi-scenario formulations, in terms of both computational requirements and accuracy of modeling the effect of uncertainty. These are likely related to the number of uncertain parameters and the level of discretization required to accurately model a specific probability distribution.

Future research here could also focus on improving the computational performance of the dynamic optimization approach. As mentioned in the case studies, gPROMS [204] performs dynamic optimization reliably, but cannot initialize the flowsheet at each iteration using the results of the previous iteration (as it can for steady-state optimization). Building such capability could dramatically decrease the amount of computational time required to solve the reformulated dynamic optimization problems. Furthermore, it may be worth investigating how the quasi-steady-state scheme presented in Part I of this dissertation can accelerate the solution of the "smoothing" approach used to treat the process flowsheet equality constraints. For instance, it may often be necessary to only treat the

slowest pseudo-transient equations as differential equations; the remaining equations may usually remain close to the inertial manifold defining their solution.

17.3 Optimization of Dynamic Process Operations

Part III of this dissertation presented concepts related to the dynamic optimization of process operations, pertaining to cyclic processes and production scheduling. Towards the former, I presented a novel methodology for modeling, simulation, and optimization of dynamic process models, with a particular focus on periodic processes. Central to the approach is a two-step reformulation of the dynamic model, consisting of a full discretization of the time and spatial domains, followed by recasting the resulting system as a differential-algebraic equation system with tunable dynamics. I characterized the stability properties of this class of systems and demonstrated its effectiveness on several case studies, including an industrial hydrogen production process. Towards the latter, I presented a data-driven approach to learn scheduling-relevant process dynamics from operating data, in the form of either physical or latent variables. With these models, the scheduling calculation is formulated as a dynamic optimization problem that is easily tractable from a computational point of view. The value of the proposed scheduling framework was demonstrated by considering the demand response operation of both a real-world and a simulated air separation unit.

Future research in periodic process optimization using pseudo-transient continuation should investigate how the stability and convergence of the method is affected in the case of mesh refinement. Many chemical engineering models present steep fronts in discretized domains, and integration of (adaptive) mesh refinement into the pseudo-transient approach may aid in numerical accuracy and stability. It may also be worth quantifying the benefits of merging steady-state and dynamic models into a single large-scale process flowsheet. Additional examples of processes with periodic steps may further reveal the impact of con-

sidering trade-offs between the (time-averaged) performance of dynamic components and the steady-state components. I note that these models may benefit from the aforementioned developments in custom or stand-alone software for pseudo-transient flowsheets.

In terms of production scheduling, future research should investigate how optimization under uncertainty affects the (expected) economic performance of optimal schedules, given inaccuracy in price, product demand, or disturbance forecasts. The results in this dissertation revealed that price forecasts have a sizable affect on the cost of a particular schedule, and I expect that considering uncertainties during scheduling will partially mitigate this effect. In a similar vein, price forecasts, process states, and resulting schedules should be updated in a moving-horizon fashion, and methods are required for such "online" applications. Additionally, other production scheduling applications should be explored, such as a refinery with changing feedstocks, multiple products, and engagements in fast-changing product markets. Finally, future works could investigate the potential computational benefits of using a latent-variable process representation in scheduling calculations when a simultaneous solver is used (i.e., the time domain is discretized to create a large-scale algebraic system).

Bibliography

[1] M Abadi, P Barham, J Chen, Z Chen, A Davis, J Dean, M Devin, S Ghemawat, G Irving, M Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, pages 265–283, 2016.

[2] A Agarwal, LT Biegler, and SE Zitney. Simulation and optimization of pressure swing adsorption systems using reduced-order modeling. *Industrial & Engineering Chemistry Research*, 48(5):2327–2343, 2008.

[3] A Agarwal, LT Biegler, and SE Zitney. A superstructure-based optimal synthesis of PSA cycles for post-combustion CO2 capture. *AIChE Journal*, 56(7):1813–1828, 2010.

[4] A Alhajaj, N Mac Dowell, and N Shah. A techno-economic analysis of post-combustion CO_2 capture and compression applied to a combined cycle gas turbine: Part II. identifying the cost-optimal control and design variables. *International Journal of Greenhouse Gas Control*, 52:331–343, 2016.

[5] E Alpay. *Rapid pressure swing adsorption processes.* PhD thesis, University of Cambridge, 1992.

[6] VS Anishchenko, T Vadivasova, and G Strelkova. *Deterministic Nonlinear Systems*. Springer: New York, NY, 2014.

[7] Elvira MB Aske, S Strand, and S Skogestad. Coordinator MPC for maximizing plant throughput. *Computers & Chemical Engineering*, 32(1-2):195–204, 2008.

[8] A Aspelund, T Gundersen, J Myklebust, MP Nowak, and A Tomasgard. An optimization-simulation model for a simple LNG process. *Computers & Chemical Engineering*, 34(10):1606–1617, 2010.

[9] AspenTech, Bedford, MA, US. AspenPlus V8.8. `www.aspentech.com/en/products/engineering/ aspen-plus`, Accessed 10/2018.

[10] G Awanou. Pseudo transient continuation and time marching methods for Monge-Ampère type equations. *Advances in Computational Mathematics*, 41(4):907–935, 2015.

[11] TA Badgwell and SJ Qin. Model-predictive control in practice. *Encyclopedia of Systems and Control*, pages 756–760, 2015.

[12] SS Bahakim and LA Ricardez-Sandoval. Optimal design of a postcombustion CO2 capture pilot-scale plant under process uncertainty: A ranking-based approach. *Industrial & Engineering Chemistry Research*, 54(15):3879–3892, 2015.

[13] S Balakrishna and LT Biegler. Targeting strategies for the synthesis and energy integration of nonisothermal reactor networks. *Industrial & Engineering Chemistry Research*, 31(9):2152–2164, 1992.

[14] M Baldea. From process integration to process intensification. *Computers & Chemical Engineering*, 81:104–114, 2015.

[15] M Baldea and P Daoutidis. *Dynamics and nonlinear control of integrated process systems*. Cambridge University Press, Cambridge, UK, 2012.

[16] M Baldea, J Du, J Park, and I Harjunkoski. Integrated production scheduling and model predictive control of continuous processes. *AIChE Journal*, 61(12):4179–4190, 2015.

[17] M Baldea and TF Edgar. Dynamic process intensification. *Current Opinion in Chemical Engineering*, 22:48–53, 2018.

[18] M Baldea and I Harjunkoski. Integrated production scheduling and process control: A systematic review. *Computers & Chemical Engineering*, 71:377–390, 2014.

[19] RW Barnard, K Pearce, and L Schovanec. Inequalities for the perimeter of an ellipse. *Journal of Mathematical Analysis and Applications*, 260(2):295–306, 2001.

[20] NP Basán, IE Grossmann, A Gopalakrishnan, I Lotero, and CA Méndez. Novel MILP scheduling model for power-intensive processes under time-sensitive electricity prices. *Industrial & Engineering Chemistry Research*, 57(5):1581–1592, 2018.

[21] J Bausa and W Marquardt. Quick and reliable phase stability test in VLLE flash calculations by homotopy continuation. *Computers & Chemical Engineering*, 24(11):2447–2456, 2000.

[22] LDR Beal, D Petersen, D Grimsman, S Warnick, and JD Hedengren. Integrated scheduling and control in discrete-time with dynamic parameters and constraints. *Computers & Chemical Engineering*, 115:361–376, 2018.

[23] LDR Beal, D Petersen, G Pila, B Davis, S Warnick, and JD Hedengren. Economic benefit from progressive integration of scheduling and control for continuous chemical processes. *Processes*, 5(4):84, 2017.

[24] G Bercic and J Levec. Intrinsic and global reaction rate of methanol dehydration over. gamma.-alumina pellets. *Ind. Eng. Chem. Res.*, 31(4):1035–1040, 1992.

[25] G Bercic and J Levec. Catalytic dehydration of methanol to dimethyl ether. Kinetic investigation and reactor simulation. *Industrial & Engineering Chemistry Research*, 32(11):2478–2484, 1993.

[26] TL Bergman, FP Incropera, DP DeWitt, and AS Lavine. *Fundamentals of heat and mass transfer*. John Wiley & Sons, 2011.

[27] LT Biegler. Solution of dynamic optimization problems by successive quadratic programming and orthogonal collocation. Technical report, DTIC Document, 1983.

[28] LT Biegler. An overview of simultaneous strategies for dynamic optimization. *Chemical Engineering and Processing: Process Intensification*, 46(11):1043–1053, 2007.

[29] LT Biegler. *Nonlinear programming: concepts, algorithms, and applications to chemical processes*. SIAM, Philadelphia, PA (United States), 2010.

[30] LT Biegler, AM Cervantes, and A Wächter. Advances in simultaneous strategies for dynamic process optimization. *Chemical Engineering Science*, 57(4):575–593, 2002.

[31] LT Biegler and IE Grossmann. Retrospective on optimization. *Computers & Chemical Engineering*, 28(8):1169–1192, 2004.

[32] LT Biegler, IE Grossmann, and AW Westerberg. *Systematic methods for chemical process design*. Prentice Hall, Upper Saddle River, NJ (United States), 1997.

[33] RB Bird, EN Lightfoot, and EW Stewart. *Transport Phenomena*. Wiley, New York, NY, 2007.

[34] D Bongartz and A Mitsos. Deterministic global flowsheet optimization: Between equation-oriented and sequential-modular methods. *AIChE Journal*, 65(3):1022–1034, 2019.

[35] F Boukouvala and CA Floudas. ARGONAUT: AlgoRithms for Global Optimization of coNstrAined grey-box compUTational problems. *Optimization Letters*, 11(5):895–913, 2017.

[36] LD Boyko and GN Kruzhilin. Heat transfer and hydraulic resistance during condensation of steam in a horizontal tube and in a bundle of tubes. *International Journal of Heat and Mass Transfer*, 10(3):361–373, 1967.

[37] KE Brenan, SL Campbell, and LR Petzold. *Numerical solution of initial-value problems in differential-algebraic equations*, volume 14. SIAM, Philadelphia, PA (United States), 1996.

[38] M Bui, CS Adjiman, A Bardow, EJ Anthony, A Boston, S Brown, PS Fennell, S Fuss, A Galindo, LA Hackett, JP Hallett, HJ Herzog, G Jackson, J Kemper, S Krevor, GC Maitland, M Matuszewski, IS Metcalfe, C Petit, G Puxty, J Reimer, DM Reiner, ES Rubin, SA Scott, N Shah, B Smit, JPM Trusler, P Webley, J Wilcox, and N Mac Dowell. Carbon capture and storage (CCS): The way forward. *Energy and Environmental Science*, 11:1062–1176, 2018.

[39] J Bukowski, YN Liu, S Boccella, and L Kowalski. Innovations in natural gas liquefaction technology for future LNG plants and floating LNG facilities. In *International Gas Union Research Conference*, 2011.

[40] J Burger, V Papaioannou, S Gopinath, G Jackson, A Galindo, and CS Adjiman. A hi-

erarchical method to integrated solvent and process design of physical CO_2 absorption using the SAFT-γ Mie approach. *AIChE Journal*, 61(10):3249–3269, 2015.

[41] B Burnak, J Katz, NA Diangelakis, and EN Pistikopoulos. Simultaneous process scheduling and control: A multiparametric programming-based approach. *Industrial & Engineering Chemistry Research*, 57(11):3963–3976, 2018.

[42] JA Caballero and IE Grossmann. An algorithm for the use of surrogate models in modular flowsheet optimization. *AIChE Journal*, 54(10):2633–2650, 2008.

[43] Y Cao, CLE Swartz, and M Baldea. Design for dynamic performance: Application to an air separation unit. In *American Control Conference (ACC), 2011*, pages 2683–2688. IEEE, 2011. San Francisco, CA.

[44] Y Cao, CLE Swartz, M Baldea, and S Blouin. Optimization-based assessment of design limitations to air separation plant agility in demand response scenarios. *Journal of Process Control*, 33:37–48, 2015.

[45] Y Cao, CLE Swartz, and J Flores-Cerrillo. Optimal dynamic operation of a high-purity air separation plant under varying market conditions. *Industrial & Engineering Chemistry Research*, 55(37):9956–9970, 2016.

[46] Y Cao, CLE Swartz, J Flores-Cerrillo, and J Ma. Dynamic modeling and collocation-based model reduction of cryogenic air separation units. *AIChE Journal*, 62(5):1602–1615, 2016.

[47] JR Cash and AH Karp. A variable order Runge-Kutta method for initial value problems with rapidly varying right-hand sides. *ACM Transactions on Mathematical Software (TOMS)*, 16(3):201–222, 1990.

[48] A Caspari, JMM Faust, Pl Schäfer, A Mhamdi, and A Mitsos. Economic nonlinear model predictive control for flexible operation of air separation units. *IFAC-PapersOnLine*, 51(20):295–300, 2018.

[49] FE Cellier and E Kofman. *Continuous system simulation*. Springer Science & Business Media, New York, NY (United States), 2006.

[50] IM Cerrillo-Briones and LA Ricardez-Sandoval. Robust optimization of a post-combustion CO_2 capture absorber column under process uncertainty. *Chemical Engineering Research and Design*, 144:386–396, 2019.

[51] M Ceze and KJ Fidkowski. Pseudo-transient continuation, solution update methods, and CFL strategies for DG discretizations of the RANS-SA equations. In *21st AIAA Computational Fluid Dynamics Conference*, page 2686, 2013.

[52] VM Charitopoulos, LG Papageorgiou, and V Dua. Closed-loop integration of planning, scheduling and multi-parametric nonlinear control. *Computers & Chemical Engineering*, 2018.

[53] E Chen, T Madan, D Sachde, MS Walters, P Nielsen, and GT Rochelle. Pilot plant results with piperazine. *Energy Procedia*, 37:1572–1583, 2013.

[54] Q Chen and IE Grossmann. Recent developments and challenges in optimization-based process synthesis. *Annual Review of Chemical and Biomolecular Engineering*, 8:249–283, 2017.

[55] TS Coffey, CT Kelley, and DE Keyes. Pseudo-transient continuation and differential-algebraic equations. *SIAM Journal on Scientific Computing*, 25(2):553–569, 2003.

[56] AJ Conejo, FJ Nogales, and JM Arroyo. Price-taker bidding strategy under price uncertainty. *IEEE Transactions on Power Systems*, 17(4):1081–1088, 2002.

[57] RG Cooper. Perspective: The Stage-Gate® idea-to-launch process—-Update, what's new, and NexGen systems. *Journal of Product Innovation Management*, 25(3):213–232, 2008.

[58] P Cruz, JC Santos, FD Magalhães, and A Mendes. Simulation of separation processes using finite volume method. *Computers & Chemical Engineering*, 30(1):83–98, 2005.

[59] F Cui, C Cui, and J Sun. Simultaneous optimization of heat-integrated extractive distillation with a recycle feed using pseudo transient continuation models. *Industrial & Engineering Chemistry Research*, 57(45):15423–15436, 2018.

[60] T Damartzis, AI Papadopoulos, and P Seferlis. Process flowsheet design optimization for various amine-based solvents in post-combustion CO_2 capture plants. *Journal of Cleaner Production*, 111:204–216, 2016.

[61] P Daoutidis, JH Lee, I Harjunkoski, S Skogestad, M Baldea, and C Georgakis. Integrating operations and control: A perspective and roadmap for future research. *Computers & Chemical Engineering*, 115:179–184, 2018.

[62] A de Klerk. Fischer–Tropsch fuels refinery design. *Energy & Environmental Science*, 4(4):1177–1205, 2011.

[63] F Del Nogal, JK Kim, S Perry, and R Smith. Optimal design of mixed refrigerant cycles. *Industrial & Engineering Chemistry Research*, 47(22):8724–8740, 2008.

[64] FA Demneh and A Mesbah. The effect of kinetic energy change on flow in gas pipelines: The significance of the acceleration term in pressure drop calculations is investigated. *Hydrocarbon Processing*, 87(5):81–84, 2008.

[65] LS Dias and MG Ierapetritou. From process control to supply chain management: An overview of integrated decision making strategies. *Computers & Chemical Engineering*, 106:826–835, 2017.

[66] LS Dias and MG Ierapetritou. Optimal operation and control of intensified processes—challenges and opportunities. *Current Opinion in Chemical Engineering*, 2019. doi:10.1016/j.coche.2018.12.008.

[67] LS Dias, RC Pattison, C Tsay, M Baldea, and MG Ierapetritou. A simulation-based optimization framework for integrating scheduling and model predictive control, and its application to air separation units. *Computers & Chemical Engineering*, 113:139–151, 2018.

[68] Y Ding and MD LeVan. Periodic states of adsorption cycles III. Convergence acceleration for direct determination. *Chemical Engineering Science*, 56(17):5217–5230, 2001.

[69] MM Donahue, BJ Roach, JJ Downs, T Blevins, M Baldea, and RB Eldridge. Dividing wall column control: common practices and key findings. *Chemical Engineering and Processing: Process Intensification*, 107:106–115, 2016.

[70] AW Dowling and LT Biegler. Rigorous optimization-based synthesis of distillation cascades without integer variables. In *24th European Symposium on Computer Aided Process Engineering*, volume 33, page 55. Elsevier, 2014.

[71] AW Dowling and LT Biegler. A framework for efficient large scale equation-oriented flowsheet optimization. *Computers & Chemical Engineering*, 72:3–20, 2015.

[72] AW Dowling, JP Eason, J Ma, DC Miller, and LT Biegler. Equation-based design, integration, and optimization of oxycombustion power systems. In *Alternative Energy Sources and Technologies*, pages 119–158. Springer, 2016.

[73] AW Dowling and VM Kumar, R Zavala. A multi-scale optimization framework for electricity market participation. *Applied Energy*, 190:147–164, 2017.

[74] J Du, J Park, I Harjunkoski, and M Baldea. A time scale-bridging approach for integrating production scheduling and process control. *Computers & Chemical Engineering*, 79:59–69, 2015.

[75] K Dubin. Petra Nova is one of two carbon capture and sequestration power plants in the world. *US Energy Information Administration: Today In Energy*, 2017. `https://www.eia.gov/todayinenergy/detail.php?id=33552`, Accessed 10/2018.

[76] MA Duran and IE Grossmann. Simultaneous optimization and heat integration of chemical processes. *AIChE Journal*, 32(1):123–138, 1986.

[77] DC Dyson and JM Simon. Kinetic expression with diffusion correction for ammonia synthesis on industrial catalyst. *Industrial & Engineering Chemistry Fundamentals*, 7(4):605–610, 1968.

[78] TF Edgar, DM Himmelblau, and LS Lasdon. *Optimization of chemical processes*. McGraw-Hill, New York, NY (United States), 2001.

[79] TF Edgar, BA Ogunnaike, JJ Downs, KR Muske, and BW Bequette. Renovating the undergraduate process control course. *Computers & Chemical Engineering*, 30(10):1749–1762, 2006.

[80] TF Edgar, BA Ogunnaike, and KR Muske. A global view of graduate process control education. *Computers & Chemical Engineering*, 30(10):1763–1774, 2006.

[81] PJ Edge, PJ Heggs, M Pourkashanian, and PL Stephenson. Integrated fluid dynamics-process modelling of a coal-fired power plant with carbon capture. *Applied Thermal Engineering*, 60(1):242–250, 2013.

[82] PJ Edge, PJ Heggs, M Pourkashanian, PL Stephenson, and A Williams. A reduced order full plant model for oxyfuel combustion. *Fuel*, 101:234–243, 2012.

[83] SSEH Elnashaie, ME Abashar, and AS Al-Ubaid. Simulation and optimization of an industrial ammonia reactor. *Industrial & Engineering Chemistry Research*, 27(11):2015–2022, 1988.

[84] SSEH Elnashaie, AT Mahfouz, and SS Elshishini. Digital simulation of an industrial ammonia reactor. *Chemical Engineering and Processing: Process Intensification*, 23(3):165–177, 1988.

[85] S Ergun. Fluid flow through packed columns. *Chemical Engineering Progress*, 48:89–94, 1952.

[86] RD Evans and LA Ricardez-Sandoval. Multi-scenario modelling of uncertainty in stochastic chemical systems. *Journal of Computational Physics*, 273:374–392, 2014.

[87] RW Fahien and JM Smith. Mass transfer in packed beds. *AIChE Journal*, 1(1):28–37, 1955.

[88] Y Fei, S Black, J Szuhánszki, L Ma, DB Ingham, PJ Stanger, and M Pourkashanian. Evaluation of the potential of retrofitting a coal power plant to oxy-firing using CFD and process co-simulation. *Fuel Processing Technology*, 131:45–58, 2015.

[89] J Flores-Cerrillo and JF MacGregor. Latent variable MPC for trajectory tracking in batch processes. *Journal of Process Control*, 15(6):651–663, 2005.

[90] A Flores-Tlacuahuac and IE Grossmann. Simultaneous cyclic scheduling and control of a multiproduct CSTR. *Industrial & Engineering Chemistry Research*, 45(20):6698–6712, 2006.

[91] CA Floudas, ZH Gümüs, and MG Ierapetritou. Global optimization in design under uncertainty: feasibility test and flexibility index problems. *Industrial & Engineering Chemistry Research*, 40(20):4267–4282, 2001.

[92] PT Frailie. *Modeling of carbon dioxide absorption/stripping by aqueous methyldiethanolamine/piperazine*. PhD thesis, The University of Texas at Austin, 2014.

[93] PT Frailie, T Madan, BJ Sherman, and GT Rochelle. Energy performance of advanced stripper configurations. *Energy Procedia*, 37:1696–1705, 2013.

[94] B Freeman, P Hao, R Baker, J Kniep, E Chen, J Ding, Y Zhang, and GT Rochelle. Hybrid membrane-absorption CO_2 capture process. *Energy Procedia*, 63:605–613, 2014.

[95] EN Fuller, PD Schettler, and JC Giddings. New method for prediction of binary gas-phase diffusion coefficients. *Industrial & Engineering Chemistry*, 58(5):18–27, 1966.

[96] J Gmehling, U Onken, W Arlt, P Grenzheuser, U Weidlich, B Kolbe, and J Rarey. *Vapor-Liquid Equilibrium Data Collection*, volume 1, Chemistry Data Series. DECHEMA, Frankfurt, 1986.

[97] M Golshadi, R Mosayebi Behbahani, and MR Irani. CFD simulation of dimethyl ether synthesis from methanol in an adiabatic fixed-bed reactor. *Iranian Journal of Oil & Gas Science and Technology*, 2(2):50–64, 2013.

[98] I Goodfellow, Y Bengio, and A Courville. *Deep learning*. MIT Press, Cambridge, MA, 2016.

[99] JT Gravdahl and O Egeland. Centrifugal compressor surge and speed control. *IEEE Transactions on Control Systems Technology*, 7(5):567–579, 1999.

[100] IE Grossmann, RM Apap, BA Calfa, P Garcia-Herreros, and Q Zhang. Recent advances in mathematical programming techniques for the optimization of process systems under uncertainty. *Computers & Chemical Engineering*, 91:3–14, 2016.

[101] IE Grossmann and KP Halemane. Decomposition strategy for designing flexible chemical plants. *AIChE Journal*, 28(4):686–694, 1982.

[102] IE Grossmann and RWH Sargent. Optimum design of heat exchanger networks. *Computers & Chemical Engineering*, 2(1):1–7, 1978.

[103] D Gupta, C Maravelias, and JM Wassick. From rescheduling to online scheduling. *Chemical Engineering Research and Design*, 116:83–97, 2016.

[104] K Gustafsson. Control theoretic techniques for stepsize selection in explicit Runge-Kutta methods. *ACM Transactions on Mathematical Software (TOMS)*, 17(4):533–554, 1991.

[105] RF Gutierrez. *A model-based methodology for managing technological risk*. PhD thesis, Imperial College London (University of London), 2007.

[106] RF Gutierrez, CC Pantelides, and CS Adjiman. Risk analysis and robust design under technological uncertainty. *Computer Aided Chemical Engineering*, 21:191–196, 2006.

[107] J Hahn and TF Edgar. An improved method for nonlinear model reduction using balancing of empirical gramians. *Computers & Chemical Engineering*, 26(10):1379–1397, 2002.

[108] MM Hasan, IA Karimi, HE Alfadala, and H Grootjans. Operational modeling of multistream heat exchangers with phase changes. *AIChE Journal*, 55(1):150–171, 2009.

[109] MMF Hasan, RC Baliban, JA Elia, and CA Floudas. Modeling, simulation, and optimization of postcombustion CO_2 capture for variable feed concentration and flow rate. 1. Chemical absorption and membrane processes. *Industrial & Engineering Chemistry Research*, 51(48):15642–15664, 2012.

[110] MMF Hasan, EL First, F Boukouvala, and CA Floudas. A multi-scale framework for CO_2 capture, utilization, and sequestration: CCUS and CCU. *Computers & Chemical Engineering*, 81:2–21, 2015.

[111] MT Heath. *Scientific Computing: An Introductory Survey*. McGraw-Hill, New York, NY, 2002.

[112] GF Hewitt. *Heat exchanger design handbook, 1998*. Begell House, Danbury, CT, 1998.

[113] DV Hinkley. On the ratio of two correlated normal random variables. *Biometrika*, 56(3):635–639, 1969.

[114] GE Hinton and RR Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

[115] K Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4(2):251–257, 1991.

[116] MIT Energy Initiative et al. The future of natural gas: An interdisciplinary MIT study. *Massachusetts Institute of Technology, Cambridge, MA*, 2011.

[117] F Jalali and JD Seader. Homotopy continuation method in multi-phase multi-reaction equilibrium systems. *Computers & Chemical Engineering*, 23(9):1319–1331, 1999.

[118] JB Jensen. *Optimal operation of refrigeration cycles*. PhD thesis, Norwegian University of Science and Technology, 2008.

[119] JB Jensen and SS. Problems with specifying ΔT_{min} in the design of processes with heat exchangers. *Industrial & Engineering Chemistry Research*, 47(9):3071–3075, 2008.

[120] L Jiang, LT Biegler, and VG Fox. Simulation and optimization of pressure-swing adsorption systems for air separation. *AIChE Journal*, 49(5):1140–1157, 2003.

[121] T Johansson. Integrated scheduling and control of an air separation unit subject to time-varying electricity prices. Master's thesis, KTH Royal Institute of Technology, Stockholm, Sweden, 2015.

[122] RS Kamath, LT Biegler, and IE Grossmann. Modeling multistream heat exchangers with and without phase changes for simultaneous optimization and heat integration. *AIChE Journal*, 58(1):190–204, 2012.

[123] CA Kang, AR Brandt, and LJ Durlofsky. Optimizing heat integration in a flexible coal–natural gas power station with CO_2 capture. *International Journal of Greenhouse Gas Control*, 31:138–152, 2014.

[124] M Karimi, M Hillestad, and HF Svendsen. Positive and negative effects on energy consumption by inter–heating of stripper in CO_2 capture plant. *Energy Procedia*, 23:15–22, 2012.

[125] D Karlstrom and P Runeson. Combining agile methods with stage-gate project management. *IEEE software*, 22(3):43–49, 2005.

[126] Y Kawajiri and LT Biegler. Optimization strategies for simulated moving bed and PowerFeed processes. *AIChE Journal*, 52(4):1343–1350, 2006.

[127] CT Kelley. *Iterative methods for optimization*, volume 18, Frontiers in Applied Mathematics. SIAM, Philadephia, PA, 1999.

[128] CT Kelley and DE Keyes. Convergence analysis of pseudo-transient continuation. *SIAM Journal on Numerical Analysis*, 35(2):508–523, 1998.

[129] MT Kelley, RC Pattison, R Baldick, and M Baldea. An MILP framework for optimizing demand response operation of air separation units. *Applied Energy*, 222:951–966, 2018.

[130] Ioannis G Kevrekidis, C William Gear, James M Hyman, Panagiotis G Kevrekidis, Olof Runborg, Constantinos Theodoropoulos, et al. Equation-free, coarse-grained multiscale computation: Enabling mocroscopic simulators to perform system-level analysis. *Communications in Mathematical Sciences*, 1(4):715–762, 2003.

[131] H Khajuria and EN Pistikopoulos. Optimization and control of pressure swing adsorption processes under uncertainty. *AIChE Journal*, 59(1):120–131, 2013.

[132] MS Khan and M Lee. Design optimization of single mixed refrigerant natural gas liquefaction process using the particle swarm paradigm with nonlinear constraints. *Energy*, 49:146–155, 2013.

[133] DP Kingma and J Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[134] AA Kiss and JPCS David. Innovative dimethyl ether synthesis in a reactive dividing-wall column. *Computers & Chemical Engineering*, 38:74–81, 2012.

[135] AA Kiss and RM Ignat. Revamping dimethyl ether separation to a single-step process. *Chemical Engineering & Technology*, 36(7):1261–1267, 2013.

[136] F Klus, Sand Nüske, P Koltai, H Wu, IG Kevrekidis, C Schütte, and F Noé. Data-driven model reduction and transfer operator approximation. *Journal of Nonlinear Science*, 28(3):985–1010, 2018.

[137] P Kokotovic, HK Khalil, and J O'Reilly. *Singular perturbation methods in control: analysis and design*, volume 25, Classics in Applied Mathematics. SIAM, Philadelphia, PA, 1999.

[138] RW Koller, LA Ricardez-Sandoval, and LT Biegler. Stochastic back-off algorithm for simultaneous design, control, and scheduling of multiproduct systems under uncertainty. *AIChE Journal*, 64(7):2379–2389, 2018.

[139] P Kou, D Liang, L Gao, and J Lou. Probabilistic electricity price forecasting with variational heteroscedastic Gaussian process and active learning. *Energy Conversion and Management*, 89:298–308, 2015.

[140] DE Kowler and RH Kadlec. The optimal control of a periodic adsorber. *AIChE Journal*, 18(6):1207–1219, 1972.

[141] MA Kramer. Nonlinear principal component analysis using autoassociative neural networks. *AIChE Journal*, 37(2):233–243, 1991.

[142] A Kumar, M Baldea, and TF Edgar. A physics-based model for industrial steam-methane reformer optimization with non-uniform temperature field. *Computers & Chemical Engineering*, 105:224–236, 2017.

[143] A Kumar, TF Edgar, and M Baldea. Multi-resolution model of an industrial hydrogen plant for plantwide operational optimization with non-uniform steam-methane reformer temperature field. *Computers & Chemical Engineering*, 107:271–283, 2017.

[144] D Kuzmin. Introduction to computational fluid dynamics. *University of Dortmund, Dortmund*, 2014.

[145] GS Ladde and DD Šiljak. Multiparameter singular perturbations of linear systems with multiple time scales. *Automatica*, 19(4):385–394, 1983.

[146] T Lafitte, V Papaioannou, S Dufal, and CC Pantelides. gSAFT: Advanced physical property prediction for process modelling. *Computer Aided Chemical Engineering*, 40:1003–1008, 2017.

[147] DA Latham, KB McAuley, BA Peppley, and TM Raybold. Mathematical modeling of an industrial steam-methane reformer for on-line deployment. *Fuel Processing Technology*, 92(8):1574–1586, 2011.

[148] D Laurí, JA Rossiter, J Sanchis, and M Martínez. Data-driven latent-variable model-based predictive control for continuous processes. *Journal of Process Control*, 20(10):1207–1219, 2010.

[149] GC Lee, R Smith, and XX Zhu. Optimal synthesis of mixed-refrigerant systems for low-temperature processes. *Industrial & Engineering Chemistry Research*, 41(20):5016–5028, 2002.

[150] JA Lee and M Verleysen. *Nonlinear dimensionality reduction*. Springer Science & Business Media, New York, NY, 2007.

[151] CF Leibovici and J Neoschil. A solution of Rachford-Rice equations for multiphase systems. *Fluid Phase Equilibria*, 112(2):217–221, 1995.

[152] X Li, A Tomasgard, and PI Barton. Natural gas production network infrastructure development under uncertainty. *Optimization and Engineering*, 18(1):35–62, 2017.

[153] Y Lin, CR Gwaltney, and MA Stadtherr. Reliable modeling and optimization for chemical engineering applications: interval analysis approach. *Reliable Computing*, 12(6):427–450, 2006.

[154] YJ Lin, E Chen, and GT Rochelle. Pilot plant test of the advanced flash stripper for CO_2 capture. *Faraday Discussions*, 192:37–58, 2016.

[155] YJ Lin and GT Rochelle. Optimization of advanced flash stripper for CO_2 capture using piperazine. *Energy Procedia*, 63:1504–1513, 2014.

[156] J Linderoth, A Shapiro, and S Wright. The empirical behavior of sampling methods for stochastic programming. *Annals of Operations Research*, 142(1):215–241, 2006.

[157] RJ Lovelett, F Dietrich, S Lee, and IG Kevrekidis. Some manifold learning considerations towards explicit model predictive control. *arXiv preprint arXiv:1812.01173*, 2018.

[158] Y Ma, Y Luo, X Ma, T Yang, D Chen, and X Yuan. Fast algorithms for equation-oriented flowsheet simulation and optimization using pseudo-transient models. *Industrial & Engineering Chemistry Research*, 57(42):14124–14142, 2018.

[159] Y Ma, Y Luo, and X Yuan. Equation-oriented optimization of reactive distillation systems using pseudo-transient models. *Chemical Engineering Science*, 195:381–398, 2019.

[160] Y Ma, Y Luo, and X Zhang, Sand Yuan. Simultaneous optimization of complex distillation systems and heat integration using pseudo-transient continuation models. *Computers & Chemical Engineering*, 108:337–348, 2018.

[161] JF MacGregor, H Yu, SG Muñoz, and J Flores-Cerrillo. Data-based latent variable methods for process analysis, monitoring and control. *Computers & Chemical Engineering*, 29(6):1217–1223, 2005.

[162] T Madan, DH Van Wagener, E Chen, and GT Rochelle. Modeling pilot plant results for CO_2 stripping using piperazine in two stage flash. *Energy Procedia*, 37:386–399, 2013.

[163] I Malinen and J Tanskanen. Homotopy parameter bounding in increasing the robustness of homotopy continuation methods in multiplicity studies. *Computers & Chemical Engineering*, 34(11):1761–1774, 2010.

[164] CD Maranas and CA Floudas. Finding all solutions of nonlinearly constrained systems of equations. *Journal of Global Optimization*, 7(2):143–182, 1995.

[165] YS Matros and GA Bunimovich. Reverse-flow operation in fixed bed catalytic reactors. *Catalysis Reviews*, 38(1):1–68, 1996.

[166] FA Michelsen, IJ Halvorsen, BF Lund, and PE Wahl. Modeling and simulation for control of the TEALARC liquified natural gas process. *Industrial & Engineering Chemistry Research*, 49(16):7389–7397, 2010.

[167] M Minceva, LS Pais, and AE Rodrigues. Cyclic steady state of simulated moving bed processes for enantiomers separation. *Chemical Engineering and Processing: Process Intensification*, 42(2):93–104, 2003.

[168] S Misra, M Kapadi, RD Gudi, and SR. Energy-efficient production scheduling of a cryogenic air separation plant. *Industrial & Engineering Chemistry Research*, 56(15):4399–4414, 2017.

[169] S Mitra, IE Grossmann, JM Pinto, and N Arora. Optimal production planning under time-sensitive electricity prices for continuous power-intensive processes. *Computers & Chemical Engineering*, 38:171–184, 2012.

[170] S Mitra, JM Pinto, and IE Grossmann. Optimal multi-scale capacity planning for power-intensive continuous processes under time-sensitive electricity prices and demand uncertainty. Part I: Modeling. *Computers & Chemical Engineering*, 65:89–101, 2014.

[171] A Mitsos, N Asprion, CA Floudas, M Bortz, M Baldea, D Bonvin, A Caspari, and P Schäfer. Challenges in process optimization for new feedstocks and energy sources. *Computers & Chemical Engineering*, 113:209–221, 2018.

[172] S Moioli and LA Pellegrini. Improved rate-based modeling of the process of CO_2 capture with PZ solution. *Chemical Engineering Research and Design*, 93:611–620, 2015.

[173] PL Mores, JI Manassaldi, NJ Scenna, JA Caballero, MC Mussati, and SF Mussati. Optimization of the design, operating conditions, and coupling configuration of combined cycle power plants and CO_2 capture processes by minimizing the mitigation cost. *Chemical Engineering Journal*, 331:870–894, 2018.

[174] PL Mores, NJ Scenna, and SF Mussati. Post-combustion CO_2 capture process: Equilibrium stage mathematical model of the chemical absorption of CO_2 into monoethanolamine (MEA) aqueous solution. *Chemical Engineering Research and Design*, 89(9):1587–1599, 2011.

[175] LFL Moro. Process technology in the petroleum refining industry—current situation and future trends. *Computers & Chemical Engineering*, 27(8):1303–1305, 2003.

[176] RL Motard, M Shacham, and EM Rosen. Steady state chemical process simulation. *AIChE Journal*, 21(3):417–436, 1975.

[177] H Müller-Steinhagen and K Heck. A simple friction pressure drop correlation for two-phase flow in pipes. *Chemical Engineering and Processing: Process Intensification*, 20(6):297–308, 1986.

[178] LT Narraway and JD Perkins. Selection of process control structure based on linear dynamic economics. *Industrial & Engineering Chemistry Research*, 32(11):2681–2692, 1993.

[179] BO Neeraas, AO Fredheim, and B Aunan. Experimental data and model for heat transfer, in liquid falling film flow on shell-side, for spiral-wound LNG heat exchanger. *International Journal of Heat and Mass Transfer*, 47(14):3565–3572, 2004.

[180] BO Neeraas, AO Fredheim, and B Aunan. Experimental shell-side heat transfer and pressure drop in gas flow for spiral-wound LNG heat exchanger. *International Journal of Heat and Mass Transfer*, 47(2):353–361, 2004.

[181] Y Nie, LT Biegler, CM Villa, and JM Wassick. Discrete time formulation for the integration of scheduling and dynamic optimization. *Industrial & Engineering Chemistry Research*, 54(16):4303–4315, 2014.

[182] NM Nikačević, AEM Huesman, PMJ Van den Hof, and AI Stankiewicz. Opportunities and challenges for process control in process intensification. *Chemical Engineering and Processing: Process Intensification*, 52:1–15, 2012.

[183] S Nilchan and CC Pantelides. On the optimisation of periodic adsorption processes. *Adsorption*, 4(2):113–147, 1998.

[184] A Nuchitprasittichai and S Cremaschi. Optimization of CO_2 capture process with aqueous amines using response surface methodology. *Computers & Chemical Engineering*, 35(8):1521–1531, 2011.

[185] RH Nyström, R Franke, I Harjunkoski, and A Kroll. Production campaign planning including grade transition sequencing and dynamic optimization. *Computers & Chemical Engineering*, 29(10):2163–2179, 2005.

[186] A Obermeier, C Windmeier, E Esche, and JU Repke. A discrete-time scheduling model for power-intensive processes taking fatigue of equipment into consideration. *Chemical Engineering Science*, 195:904–920, 2019.

[187] SY Oh, M Binns, H Cho, and JK Kim. Energy minimization of MEA-based CO_2 capture process. *Applied Energy*, 169:353–362, 2016.

[188] JI Otashu and M Baldea. Grid-level "battery" operation of chemical processes and demand-side participation in short-term electricity markets. *Applied Energy*, 220:562–575, 2018.

[189] CC Pantelides and JG Renfro. The online use of first-principles models in process operations: Review, current status and future needs. *Computers & Chemical Engineering*, 51:136–148, 2013.

[190] RC Pattison and M Baldea. Equation-oriented flowsheet simulation and optimization using pseudo-transient models. *AIChE Journal*, 60(12):4104–4123, 2014.

[191] RC Pattison and M Baldea. Optimal design of air separation plants with variable electricity pricing. *MR Eden, JD Siirola, GP Towler, editors: Proceedings of Foundations of Computer-Aided Process Design (FOCAPD)*, pages 393–398, 2014.

[192] RC Pattison and M Baldea. Multistream heat exchangers: Equation-oriented modeling and flowsheet optimization. *AIChE Journal*, 61(6):1856–1866, 2015.

[193] RC Pattison, AM Gupta, and M Baldea. Equation-oriented optimization of process flowsheets with dividing-wall columns. *AIChE Journal*, 2015.

[194] RC Pattison, P Schmal, and CC Pantelides. Efficient computation of cyclic steady states in periodic adsorption processes using the JFNK method. In *2015 AIChE Annual Meeting*, Salt Lake City, UT, 2015. AIChE.

[195] RC Pattison, CR Touretzky, I Harjunkoski, and M Baldea. Moving horizon closed-loop production scheduling using dynamic process models. *AIChE Journal*, 63(2):639–651, 2017.

[196] RC Pattison, CR Touretzky, T Johansson, I Harjunkoski, and M Baldea. Optimal process operations in fast-changing electricity markets: framework for scheduling with low-order dynamic models and an air separation application. *Industrial & Engineering Chemistry Research*, 55(16):4562–4584, 2016.

[197] RC Pattison, C Tsay, and M Baldea. Pseudo-transient models for multiscale, multiresolution simulation and optimization of intensified reaction/separation/recycle processes: Framework and a dimethyl ether production case study. *Computers & Chemical Engineering*, 105:161–172, 2017.

[198] GE Paules and CA Floudas. Stochastic programming in process synthesis: a two-stage model with MINLP recourse for multiperiod heat-integrated distillation sequences. *Computers & Chemical Engineering*, 16(3):189–210, 1992.

[199] K Pearson. LIII. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901.

[200] FB Petlyuk, VM Platonov, and DM Slavinsky. Thermodynamically optimal method for separating multicomponent mixtures. *International Chemical Engineering*, 5(3):555, 1965.

[201] ZN Pintarič and Z Kravanja. A methodology for the synthesis of heat exchanger networks having large numbers of uncertain parameters. *Energy*, 92:373–382, 2015.

[202] EN Pistikopoulos. Perspectives in multiparametric programming and explicit model predictive control. *AIChE Journal*, 55(8):1918–1925, 2009.

[203] EN Pistikopoulos and MG Ierapetritou. Novel approach for optimal process design under uncertainty. *Computers & Chemical Engineering*, 19(10):1089–1110, 1995.

[204] Process Systems Enterprise. general PROcess Modeling System (gPROMS). `www.psenterprise.com/gproms`, 1997-2020.

[205] PC Psarras, S Comello, P Bains, P Charoensawadpong, S Reichelstein, and J Wilcox. Carbon capture and utilization in the industrial sector. *Environmental Science & Technology*, 51(19):11440–11449, 2017.

[206] SJ Qin and TA Badgwell. A survey of industrial model predictive control technology. *Control Engineering Practice*, 11(7):733–764, 2003.

[207] HN Rao and IA Karimi. A superstructure-based model for multistream heat exchanger design within flow sheet optimization. *AIChE Journal*, 63(9):3764–3777, 2017.

[208] S Recker, M Skiborowski, C Redepenning, and W Marquardt. A unifying framework for optimization-based design of integrated reaction–separation processes. *Computers & Chemical Engineering*, 81:260–271, 2015.

[209] AM Ribeiro, CA Grande, FVS Lopes, JM Loureiro, and AE Rodrigues. A parametric study of layered bed PSA for hydrogen purification. *Chemical Engineering Science*, 63(21):5258–5273, 2008.

[210] GT Rochelle. Amine scrubbing for CO_2 capture. *Science*, 325(5948):1652–1654, 2009.

[211] GT Rochelle, E Chen, S Freeman, D Van Wagener, Q Xu, and A Voice. Aqueous piperazine as the new standard for CO_2 capture technology. *Chemical Engineering Journal*, 171(3):725–733, 2011.

[212] HM Rodriguez, A Cano, M Matzopoulos, et al. Improve engineering via whole-plant design optimization: New simulation methods identify cost-effective advantages early. *Hydrocarbon Processing*, 89(12):43–49, 2010.

[213] J Rodriguez, A Andrade, A Lawal, N Samsatli, M Calado, A Ramos, T Lafitte, J Fuentes, and CC Pantelides. An integrated framework for the dynamic modelling of solvent-based CO_2 capture processes. *Energy Procedia*, 63:1206–1217, 2014.

[214] WM Rohsenow, JP Hartnett, and YI Cho. *Handbook of heat transfer*, volume 3. McGraw-Hill, New York, NY, 1998.

[215] WC Rooney and LT Biegler. Optimal process design with model parameter uncertainty and process variability. *AIChE Journal*, 49(2):438–449, 2003.

[216] LM Rose. *Distillation design in practice*. Elsevier Science Inc., New York, NY (United States), 1985.

[217] NV Sahinidis. Optimization under uncertainty: state-of-the-art and opportunities. *Computers & Chemical Engineering*, 28(6):971–983, 2004.

[218] TD Sanger. Optimal unsupervised learning in a single-layer linear feedforward neural network. *Neural networks*, 2(6):459–473, 1989.

[219] HIH Saravanamuttoo, GFC Rogers, H Cohen, and PV Straznicky. *Gas turbine theory*. Pearson Education: New York, NY, 2001.

[220] RWH Sargent. Applications of an electronic digital computer in the design of low temperature plant. *Transactions of the Institution of Chemical Engineers*, 36:201–214, 1958.

[221] RWH Sargent. Forecasts and trends in systems engineering. *The Chemical Engineer*, 262:226–230, 1972.

[222] P Schäfer, A Caspari, K Kleinhans, A Mhamdi, and A Mitsos. Reduced dynamic modeling approach for rectification columns based on compartmentalization and artificial neural networks. *AIChE Journal*, 65(5):e16568, 2019.

[223] P Schäfer, HG Westerholt, AM Schweidtmann, S Ilieva, and A Mitsos. Model-based bidding strategies on the primary balancing market for energy-intense processes. *Computers & Chemical Engineering*, 120:4–14, 2019.

[224] WE Schiesser. *The Numerical Method of Lines: Integration of Partial Differential Equations*. Academic Press (Elsevier): Cambridge, MA, 2012.

[225] CA Schnepper and MA Stadtherr. Robust process simulation using interval methods. *Computers & Chemical Engineering*, 20(2):187–199, 1996.

[226] B Schölkopf, A Smola, and KR Müller. Kernel principal component analysis. In *International Conference on Artificial Neural Networks*, pages 583–588. Springer, 1997.

[227] M Schultes. Raschig super-ring: a new fourth generation packing offers new advantages. *Chemical Engineering Research and Design*, 81(1):48–57, 2003.

[228] MA Schultz, DG Stewart, JM Harris, SP Rosenblum, MS Shakur, and DE O'Brien. Reduce costs with dividing wall columns. *Chemical Engineering Progress*, 98(5):64–71, 2002.

[229] DE Seborg, DA Mellichamp, TF Edgar, and FJ Doyle III. *Process dynamics and control, 3rd ed.* John Wiley & Sons: Hoboken, NJ, 2011.

[230] WD Seider, JD Seader, DR Lewin, and S Widagdo. *Product & design principles: Synthesis, analysis, and evaluation.* John Wiley & Sons, Hoboken, NJ (United States), 2009.

[231] WD Seider, S Widagdo, JD Seader, and DR Lewin. Perspectives on chemical product and process design. *Computers & Chemical Engineering*, 33(5):930–935, 2009.

[232] M Shacham, S Macchieto, LF Stutzman, and P Babcock. Equation oriented approach to process flowsheeting. *Computers & Chemical Engineering*, 6(2):79–95, 1982.

[233] AI Shestakov, JL Milovich, and A Noy. Solution of the nonlinear Poisson–Boltzmann equation using pseudo-transient continuation and the finite element method. *Journal of Colloid and Interface Science*, 247(1):62–79, 2002.

[234] CI Siettos, CC Pantelides, and IG Kevrekidis. Enabling dynamic process simulators to perform alternative tasks: A time-stepper-based toolkit for computer-aided analysis. *Industrial & Engineering Chemistry Research*, 42(26):6795–6801, 2003.

[235] JJ Siirola. Industrial applications of chemical process synthesis. *Advances in Chemical Engineering*, 23:1–62, 1996.

[236] PL Silveston and RR Hudgins. *Periodic Operation of Chemical Reactors.* Butterworth-Heinemann: Waltham, MA, 2012.

[237] JM Simkoff and M Baldea. Production scheduling and linear MPC: complete integration via complementarity conditions. *Computers & Chemical Engineering*, 125:287–305, 2019.

[238] AP Simpson and AE Lutz. Exergy analysis of hydrogen production via steam methane reforming. *International Journal of Hydrogen Energy*, 32(18):4811–4820, 2007.

[239] M Skiborowski. Process synthesis and design methods for process intensification. *Current Opinion in Chemical Engineering*, 22:216–225, 2018.

[240] MA Stadtherr and CM Hilton. On efficient solution of large-scale Newton-Raphson-based flowsheeting problems in limited core. *Computers & Chemical Engineering*, 6(2):115–120, 1982.

[241] G Stephenson and L Wang. Dynamic simulation of liquefied natural gas processes. *Hydrocarbon Processing: Liquefied Natural Gas Developments*, pages 37–44, 2010.

[242] LE Sterman and BE Ydstie. Periodic forcing of the CSTR: An application of the generalized Π-criterion. *AIChE Journal*, 37(7):986–996, 1991.

[243] S Strelzoff. *Technology and manufacture of ammonia*. John Wiley & Sons, 1981.

[244] Sulzer Chemtech. Structure packings: for distillation, absorption and reactive distillation. `www.sulzer.com/-/media/files/products/separation-technology/liquid_liquid_extraction/brochures/structured_packings.ashx`, Accessed 10/2018.

[245] K Tak, I Lee, H Kwon, J Kim, D Ko, and I Moon. Comparison of multistage compression configurations for single mixed refrigerant processes. *Industrial & Engineering Chemistry Research*, 54(41):9992–10000, 2015.

[246] YS Teh and GP Rangaiah. A study of equation-solving and Gibbs free energy minimization methods for phase equilibrium calculations. *Chemical Engineering Research and Design*, 80(7):745–759, 2002.

[247] LS Teske. *Integrating rate based models into a multi-objective process design & optimisation framework using surrogate models*. PhD thesis, École Polytechnique Fédérale de Lausanne, 2014.

[248] The Mathworks, Inc., Natick, MA, US. MATLAB 2016b. `www.mathworks.com/products/matlab`.

[249] Y Tian, SE Demirel, MMF Hasan, and EN Pistikopoulos. An overview of process systems engineering approaches for process intensification: State of the art. *Chemical Engineering and Processing: Process Intensification*, 133:160–210, 2018.

[250] A Toumi, F Hanisch, and S Engell. Optimal operation of continuous chromatographic processes: mathematical optimization of the VARICOL process. *Industrial & Engineering Chemistry Research*, 41(17):4328–4337, 2002.

[251] CR Touretzky and M Baldea. Nonlinear model reduction and model predictive control of residential buildings with energy recovery. *Journal of Process Control*, 24(6):723–739, 2014.

[252] RE Tsai, AF Seibert, RB Eldridge, and GT Rochelle. A dimensionless model for predicting the mass-transfer area of structured packing. *AIChE Journal*, 57(5):1173–1184, 2011.

[253] C Tsay and M Baldea. Scenario-free optimal design under uncertainty of the prico natural gas liquefaction process. *Industrial & Engineering Chemistry Research*, 57(17):5868–5880, 2018.

[254] C Tsay and M Baldea. Fast and efficient chemical process flowsheet simulation by pseudo-transient continuation on inertial manifolds. *Computer Methods in Applied Mechanics and Engineering*, 348:935–953, 2019.

[255] C Tsay and M Baldea. Integrating production scheduling and process control using latent variable dynamic models. *Control Engineering Practice*, 94:104201, 2020.

[256] C Tsay, M Baldea, J Shi, A Kumar, and J Flores-Cerrillo. Data-driven models and algorithms for demand response scheduling of air separation units. In G. P. Towler M. R. Eden, M. G. Ierapetritou, editor, *Computer Aided Chemical Engineering*, volume 44, pages 1273–1278, Proceedings of the 13th International Symposium on Process Systems Engineering, San Diego, CA, 2018. Elsevier.

[257] C Tsay, A Kumar, TF Edgar, and M Baldea. Integrating steady-state and dynamic models for multi-scale flowsheet optimization: A steam-methane reforming case study. In *Computer Aided Chemical Engineering*, volume 47, pages 403–408. Elsevier, 2019.

[258] C Tsay, A Kumar, J Fores-Cerrillo, and M Baldea. Optimal demand response scheduling of an industrial air separation unit using data-driven dynamic models. *Computers & Chemical Engineering*, 126:22–34, 2019.

[259] C Tsay, RC Pattison, and M Baldea. A dynamic optimization approach to probabilistic process design under uncertainty. *Industrial & Engineering Chemistry Research*, 56(30):8606–8621, 2017.

[260] C Tsay, RC Pattison, and M Baldea. Equation-oriented simulation and optimization of process flowsheets incorporating detailed spiral-wound multistream heat exchanged models. *AIChE Journal*, 63(9):3778–3789, 2017.

[261] C Tsay, RC Pattison, and M Baldea. A pseudo-transient optimization framework for periodic processes: Pressure swing adsorption and simulated moving bed chromatography. *AIChE Journal*, 64(8):2982–2996, 2018.

[262] C Tsay, RC Pattison, MR Piana, and M Baldea. A survey of optimal process design capabilities and practices in the chemical and petrochemical industries. *Computers & Chemical Engineering*, 112:180–189, 2018.

[263] C Tsay, RC Pattison, Y Zhang, GT Rochelle, and M Baldea. Rate-based modeling and economic optimization of next-generation amine-based carbon capture plants. *Applied Energy*, 252:113379, 2019.

[264] R Turton, RC Bailie, WB Whiting, and JA Shaeiwitz. *Analysis, synthesis and design of chemical processes*. Prentice Hall, Upper Saddle River, NJ (United States), 2008.

[265] AJV Underwood. Graphical computation of logarithmic mean temperature difference. *Industrial Chemist*, 9(167), 1933.

[266] L Van Der Maaten, E Postma, and J Van den Herik. Dimensionality reduction: a comparative review. *Journal of Machine Learning Research*, 10:66–71, 2009.

[267] DH Van Wagener and GT Rochelle. Cold rich bypass to strippers for CO_2 capture by concentrated piperazine. *Chemical Engineering & Technology*, 37(1):149–156, 2014.

[268] VS Vassiliadis, RWH Sargent, and CC Pantelides. Solution of a class of multistage dynamic optimization problems. 1. Problems without path constraints. *Industrial & Engineering Chemistry Research*, 33(9):2111–2122, 1994.

[269] VS Vassiliadis, RWH Sargent, and CC Pantelides. Solution of a class of multistage dynamic optimization problems. 2. Problems with path constraints. *Industrial & Engineering Chemistry Research*, 33(9):2123–2133, 1994.

[270] SRR Vetukuri, LT Biegler, and A Walther. An inexact trust-region algorithm for the optimization of periodic adsorption processes. *Industrial & Engineering Chemistry Research*, 49(23):12004–12013, 2010.

[271] J Viswanathan and IE Grossmann. An alternate MINLP model for finding the number of trays required for a specified separation objective. *Computers & Chemical Engineering*, 17(9):949–955, 1993.

[272] N Wakao, S Kaguei, and T Funazkri. Effect of fluid dispersion coefficients on particle-to-fluid heat transfer coefficients in packed beds: correlation of Nusselt numbers. *Chemical Engineering Science*, 34(3):325–336, 1979.

[273] MS Walters, TF Edgar, and GT Rochelle. Dynamic modeling and control of an intercooled absorber for post-combustion CO_2 capture. *Chemical Engineering and Processing: Process Intensification*, 107:1–10, 2016.

[274] MS Walters, YJ Lin, DJ Sachde, TF Edgar, and GT Rochelle. Control relevant model of amine scrubbing for CO_2 capture from power plants. *Industrial & Engineering Chemistry Research*, 55(6):1690–1700, 2016.

[275] C Wang, D Song, FA Seibert, and GT Rochelle. Dimensionless models for predicting the effective area, liquid-film, and gas-film mass-transfer coefficients of packing. *Industrial & Engineering Chemistry Research*, 55(18):5373–5384, 2016.

[276] R Wang, TF Edgar, and M Baldea. A geometric framework for monitoring and fault detection for periodic processes. *AIChE Journal*, 63(7):2719–2730, 2017.

[277] S Wang and M Baldea. Identification-based optimization of dynamical systems under uncertainty. *Computers & Chemical Engineering*, 64:138–152, 2014.

[278] HAJ Watson and PI Barton. Modeling phase changes in multistream heat exchangers. *International Journal of Heat and Mass Transfer*, 105:207–219, 2017.

[279] TL Wayburn and JD Seader. Homotopy continuation methods for computer-aided process design. *Computers & Chemical Engineering*, 11(1):7–25, 1987.

[280] A Wechsung, A Aspelund, T Gundersen, and PI Barton. Synthesis of heat exchanger networks at subambient conditions with compression and expansion of process streams. *AIChE Journal*, 57(8):2090–2108, 2011.

[281] AW Westerberg. A retrospective on design and process synthesis. *Computers & Chemical Engineering*, 28(4):447–458, 2004.

[282] AW Westerberg and TJ Berna. Decomposition of very large-scale Newton-Raphson based flowsheeting problems. *Computers & Chemical Engineering*, 2(1):61–63, 1978.

[283] J Wiebe, I Cecílio, and R Misener. Data-driven optimization of processes with degrading equipment. *Industrial & Engineering Chemistry Research*, 57(50):17177–17191, 2018.

[284] K Willcox and J Peraire. Balanced model reduction via the proper orthogonal decomposition. *AIAA Journal*, 40(11):2323–2330, 2002.

[285] TJ Williams and RE Otto. A generalized chemical processing model for the investigation of computer control. *Transactions of the American Institute of Electrical Engineers, Part I: Communication and Electronics*, 79(5):458–473, 1960.

[286] Q Xu. *Thermodynamics of CO_2 Loaded Aqueous Amines*. PhD thesis, The University of Texas at Austin, 2011.

[287] Hi Yang, C Yin, B Jiang, and D Zhang. Optimization and analysis of a VPSA process for N_2/CH_4 separation. *Separation and Purification Technology*, 134:232–240, 2014.

[288] SI Yang, DY Choi, SC Jang, SH Kim, and DK Choi. Hydrogen separation by multi-bed pressure swing adsorption of synthesis gas. *Adsorption*, 14(4):583–590, 2008.

[289] RE Young. Petroleum refining process control and real-time optimization. *IEEE control systems*, 26(6):73–83, 2006.

[290] H Zhang, A Bonilla-Petriciolet, and GP Rangaiah. A review on global optimization methods for phase equilibrium modeling and calculations. *The Open Thermodynamics Journal*, 5(S1), 2011.

[291] M Zhang and Y Guo. Rate based modeling of absorption and regeneration for CO_2 capture by aqueous ammonia solution. *Applied Energy*, 111:142–152, 2013.

[292] Q Zhang and IE Grossmann. Enterprise-wide optimization for industrial demand side management: Fundamentals, advances, and perspectives. *Chemical Engineering Research and Design*, 116:114–131, 2016.

[293] Q Zhang, IE Grossmann, CF Heuberger, A Sundaramoorthy, and JM Pinto. Air separation with cryogenic energy storage: optimal scheduling considering electric energy and reserve markets. *AIChE Journal*, 61(5):1547–1558, 2015.

[294] Q Zhang, A Sundaramoorthy, IE Grossmann, and JM Pinto. A discrete-time scheduling model for continuous power-intensive process networks with various power contracts. *Computers & Chemical Engineering*, 84:382–393, 2016.

[295] Y Zhang, H Chen, CC Chen, JM Plaza, R Dugas, and GT Rochelle. Rate-based process modeling study of CO_2 capture with aqueous monoethanolamine solution. *Industrial & Engineering Chemistry Research*, 48(20):9233–9246, 2009.

[296] Y Zhang, B Freeman, P Hao, and GT Rochelle. Absorber modeling for NGCC carbon capture with aqueous piperazine. *Faraday Discussions*, 192:459–477, 2016.

[297] Y Zhang, K Hidajat, and AK Ray. Multi-objective optimization of simulated moving bed and Varicol processes for enantio-separation of racemic pindolol. *Separation and Purification Technology*, 65(3):311–321, 2009.

[298] Y Zhang, D Sachde, E Chen, and GT Rochelle. Modeling of absorber pilot plant performance for CO_2 capture with aqueous piperazine. *International Journal of Greenhouse Gas Control*, 64:300–313, 2017.

[299] F Zhao, X Chen, and L Zhu. A hybrid numerical-symbolic solving strategy for equation-oriented process simulation and optimization. *AIChE Journal*, 63(7):2764–2780, 2017.

[300] S Zhao, MP Ochoa, L Tang, I Lotero, A Gopalakrishnan, and IE Grossmann. Novel formulation for optimal schedule with demand side management in multi-product air separation processes. *Industrial & Engineering Chemistry Research*, 58(8):3104–3117, 2019.

[301] D Zhou, K Zhou, L Zhu, J Zhao, Z Xu, Z Shao, and X Chen. Optimal scheduling of multiple sets of air separation units with frequent load-change operation. *Separation and Purification Technology*, 172:178–191, 2017.

[302] Y Zhu. *Multivariable system identification for process control*. Elsevier Science, Oxford, UK, 2001.

[303] Y Zhu, S Legg, and CD Laird. Optimal design of cryogenic air separation columns under uncertainty. *Computers & Chemical Engineering*, 34(9):1377–1384, 2010.

[304] J Zhuge and MG Ierapetritou. Integration of scheduling and control with closed loop implementation. *Industrial & Engineering Chemistry Research*, 51(25):8550–8565, 2012.

[305] J Zhuge and MG Ierapetritou. Integration of scheduling and control for batch processes using multi-parametric model predictive control. *AIChE Journal*, 60(9):3169–3183, 2014.

[306] J Zhuge and MG Ierapetritou. An integrated framework for scheduling and control using fast model predictive control. *AIChE Journal*, 61(10):3304–3319, 2015.